電子技術とコンピュータ

エレクトロニクスライフ

平成5年9月1日発行（毎月1回1日発行）
9月号733号昭和21年12月27日第三種郵便物認可

監修 NHK放送技術研究所長 泉 武博

9
1993

FAXモデム専科
多機能画像入出力システムの製作
各種プログラムの解説

ラインレベル・コンバータの製作

ディジタルオーディオ 最大値測定装置の製作

ILAスーパープロジェクタ

電波吸収体やSHF放送局が活躍

NHK放送技術研究所公開から

HiWIRE II

Victor
"HIS MASTER'S VOICE"

デジタル映像回路の状態がテレビに映る！ 映像機器修理の簡単ツール、誕生。

DS-B525 DIGITAL SCOPE
OSCILLO SCOPE
GND TP
DIGITAL IN
SYNC IN
MONITOR
MONITOR DIGITAL VIDEO
SYNC INT EXT
Victor

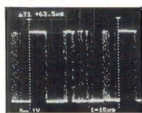
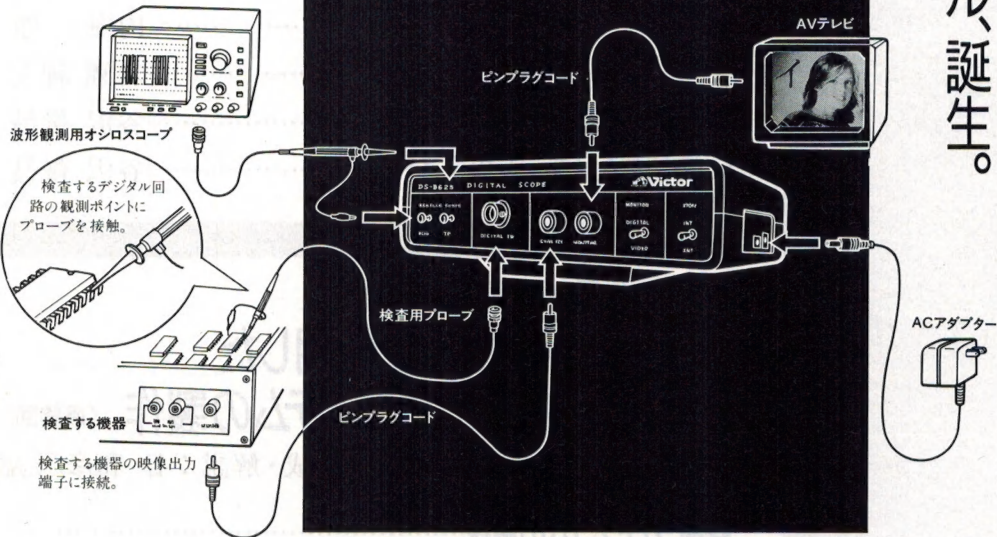
デジタル技術者待望の診断器！

オシロスコープでの波形観測では判断しにくかった、デジタル映像信号回路の良否も、これならカンタン。このデジタル・スコープ「DS-B525」が、デジタル信号をモノクロ映像信号に変換。テレビ画面に映し出される映像によって、回路の不良箇所をアナログ感覚で簡単に発見することができます。

- デジタル映像信号を画像変換して、原画と比較しながら簡単に診断可能。
- 小型軽量で携帯に便利。
- 電源は9〜12Vの各種ACアダプターに対応。
- オシロ用測定端子も装備。測定入力端子にはBNC端子を採用。

オシロスコープによるチェック

デジタル・スコープによるチェック

AVテレビ
ピンプラグコード
波形観測用オシロスコープ
検査するデジタル回路の観測ポイントにプローブを接触。
検査用プローブ
ACアダプター
検査する機器
ピンプラグコード
検査する機器の映像出力端子に接続。

デジタル・スコープ
DS-B525
本体標準価格 45,000円（税別）

《用途》
- 修理サービス
- 商品品質検査
- 教育・研修
- 開発設計等

《仕様》

デジタル入力	C-MOS 1ゲート(0〜5V)(BNC)
シンク入力	1.0Vp-p・75Ω(RCAピンジャック/コンポジット・ビデオ)
モニター出力	1.0Vp-p・75Ω(RCAピンジャック/コンポジット・ビデオ)
外形寸法	幅133×高さ44×奥行106mm
重量	約250g
オプション（別売）	検査用シンクロプローブ/ピンプラグコード/ACアダプター(9V)

製造・発売元
ビクターサービスエンジニアリング株式会社
〒113 東京都文京区本郷3-14-7

販売代理店
トーツーハン
株式会社 東通販
●お問合せは下記EL係まで
TEL. (03)3255-1051
FAX. (03)3253-3117
〒101 東京都千代田区外神田1-6-6

エレクトロニクスライフ

EL

9
1993

特集 1

FAXモデム専科

企画/構成：谷沢 俊昌

特集 2

パソコンにTV画面が取り込める

フレームメモリーを使用した 多機能画像入出力システムの製作 〈最終回〉

企画・構成・解説：谷 和彦/土屋 徹

発行――日本放送出版協会

表紙構成：道吉デザイン　表紙撮影：豊田 靖雄　青山写真スタジオ
レイアウト：橋詰 尚己／㈱キャニング／K's Labo
図版作成：新生社／㈱ノムラ　イラスト：今井 雅巳

口絵

製作

EL COLUMN

エレクトロニクス／連載

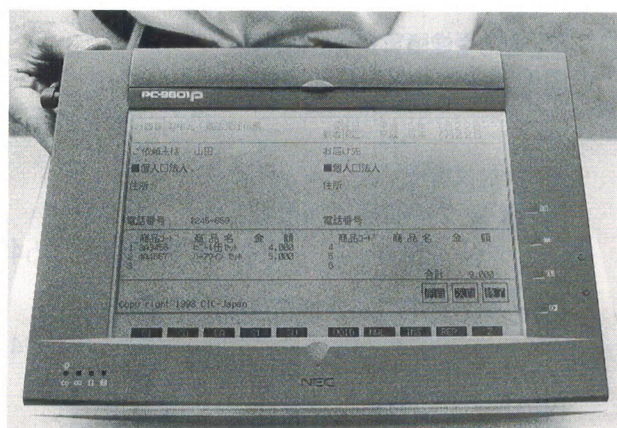
NECパソコンフェア'93

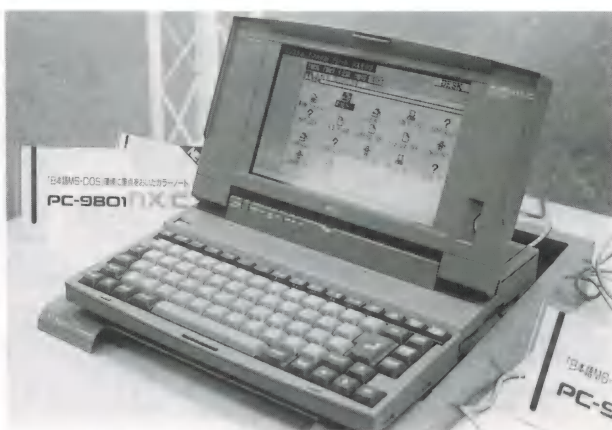
日本のウィンドウズ時代—98ファミリーとウィンドウズ対応ソフト—

去る7月22日から24日までの3日間，千葉の幕張メッセでNECの主催で,「NECパソコンフェア'93」が開催された。同フェアは今回で13回を数え，回を重ねるたびに来場者が増え，フェアも拡大しつつあるようだ。特に今回は，開催期間が夏休み中とあって，日ごろ参加できずにいた人たちも数多く見られた。

また，会場の一角でPCエンジンソフトの展示やデモもあり，家族づれでの参加も多かった。

今回は，Microsoft Windows3.1発表による「ウィンドウズ時代」に焦点を合わせ,「ウィンドウズ対応」の新製品9821Apをはじめ，ハードやソフトが多数公開・展示された。

協賛メーカーも，教育・通信

CPUに[i486]を搭載したペンパソコン「PC-9801P」。7.6MBのユーザーズメモリーや80MBの1.8インチHDDを標準内蔵。A4サイズで，重量1.6kg。最長で約6時間のバッテリ駆動が可能。あらかじめ3種類のOSをおのおの組み込んだモデルがあり，用途に応じ選択が可能。

CPUに[Pentium]を搭載した「PC-9821Af」。[9821Ap]に比べ処理速度が1.5倍と高速化し、最大で79.6MBまでメモリーの拡張が可能。[Window3.1][フルカラーウィンドウアクセラレータボードA]標準装備により、高度なグラフィック処理に対応している。

上 [日本語MS-DOS]環境に適した普及型カラーノートパソコン「PC-9801NX/C」。カードスロットルを標準装備し，拡張カードに対応。FDDモデルとHDDモデルがある。
左 [ウィンドウ環境]に最適な機能を持つ「PC-9821Ne」。256色表示，640×480/400ドットのTFTカラー液晶搭載。小型トラックボール標準装備。「PC-9801NA/C」に比べ約1.6倍の高速化している。アップグレードにも対応可能。

/ネットワーク・ビジネス・CAD・マルチメディア/グラフィック・ゲーム・周辺機器などのメーカーから，299社の出展があり，盛況でかつ「ウィンドウ対応」ソフトは人気であった。

期間中，併設のセミナー会場では，「98レベルアップセミナー」や「Windows 3.1入門セミナー」などが開催され，こちらも多数の動員をみた。

また，隣接するイベントホールでは「NECパソコンアートフェスティバル」が催され，期間中，幕張メッセはNECファミリー一色となった。

なお，同社では今回の幕張のほかに，12月には大阪・名古屋でも「NECパソコンフェア」の開催を予定しているとのこと。

（須藤）

小型ページプリンタ「PC-PR1000/4R」。解像度400dpiとSET機能により，600dpi相当の高品位印刷が可能。A4用紙を1分間に6ページ印刷可能。明朝体・ゴチック体のアウトラインフォント標準装備。

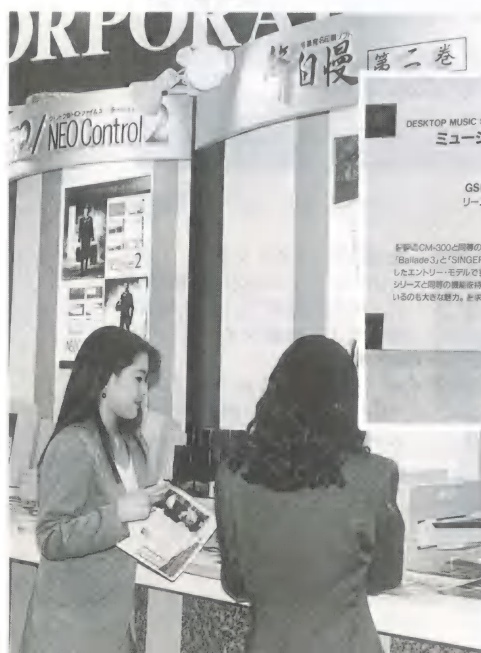
上 携帯型ファクシミリ「i300 image terminal」。3電源方式。携帯電話対応。ECM(自動誤り再送) 機能やメモリー機能を搭載。
左 テレターミナルアダプタ「PC-CM501」。2周波単信方式，出力5Wのパソコン用無線通信機。

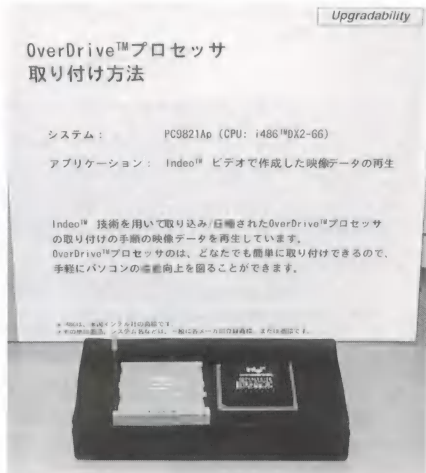
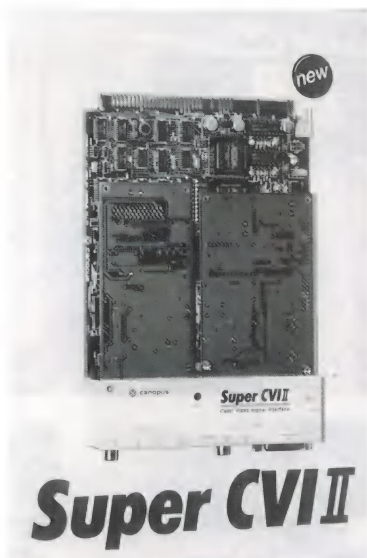
3.5インチ光ディスクユニット「PC−OD301」と5インチ光ディスクユニット「PC−OD501」。ISO規格の3.5インチ版は128Mバイト，エアフィルタ標準装備。5インチ版は片面300Mバイトの両面で600Mバイトで平均シーク時間32msの高速アクセスタイプ。

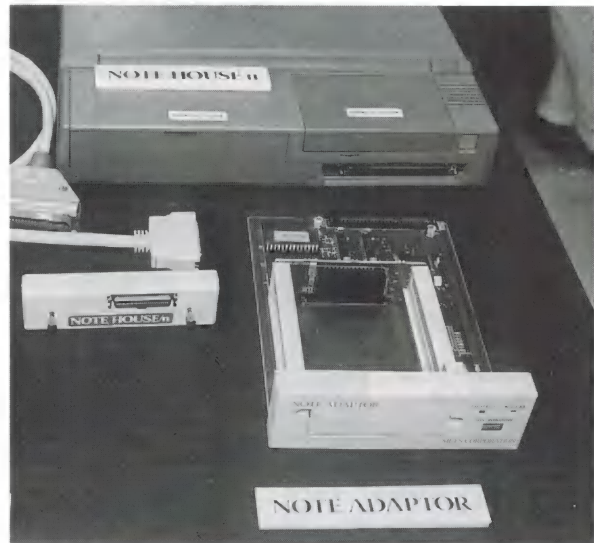
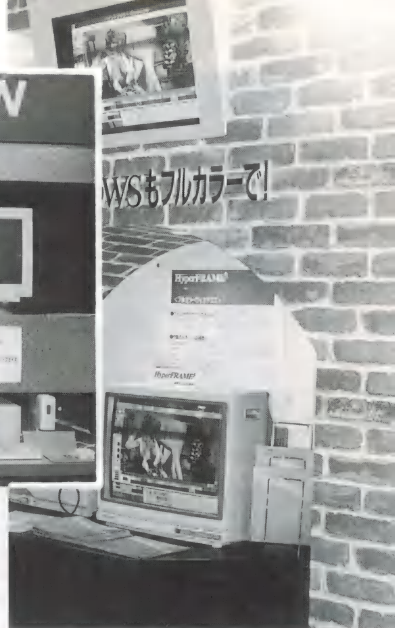
ミニカートリッジテープユニット「PC−MT150」とカセット磁気テープユニット「PC−MT600B」

コミュニケーション支援装備「トーキングパートナー」。スイッチで文章や単語・文字を選択し，発声させることでコミュニケーションをはかる。スイッチにはフットスイッチや吸気スイッチなどがある。

キーボード入力支援装置「ソフトパートナー」。タッチパネルやさまざまなスイッチを使い利用者の環境に適応するシステム。「トーキングパートナー」や「ソフトパートナー」は社会福祉の面で，コンピュータをうまく取り入れ活用する工夫がされている。

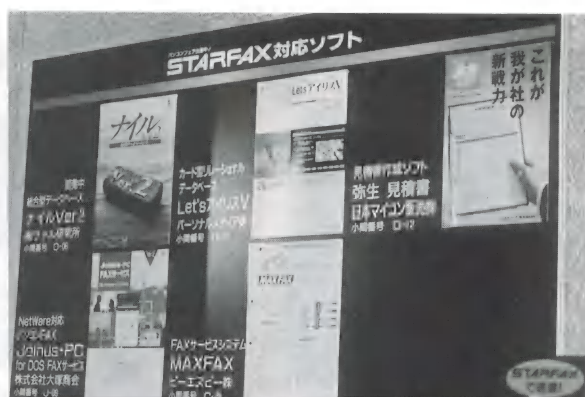
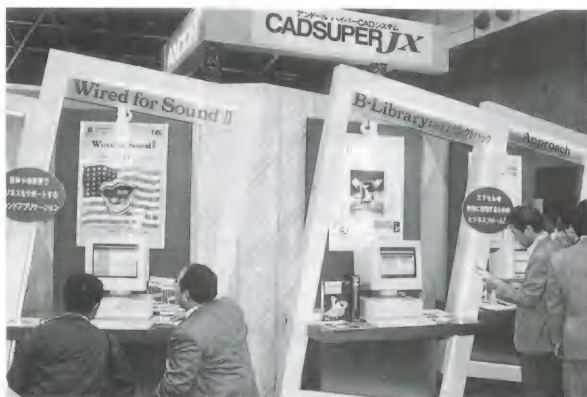
▼会場風景

ローランドのミュージックボード「ミュージ郎300BOARD V3」
右　インテルの画像用オーバードライブプロセッサ。

会場風景▶

カノープスの画像処理用ボード「Super CVI Ⅱ」。 高解像度・高画質ビデオディジタイズ・ビデオ出力ゲインロックなどの機能を持つ。

アイシーエムの17インチフラットディスプレイ「LS-2617FN」。マルチレゾリューション対応で，ノングレア処理。最大解像度1280×1024ドットの高品位低価格ディスプレイ。

インタフェース社のマルチCRTディスプレイシステム。

メッツのノートパソコン用ハードディスクアダプタ「NOTE HOUSE/n」。自動SCSI認識・両方向フォーマット／高速フォーマット機能などを持つ。

右　ビジネス関連ソフトを数多く出している，オービックビジネスコンサルタントの販売管理システム「商人」・仕入管理システム「蔵人」・給与計算ソフト「給与奉行」のパッケージのディスプレイ。

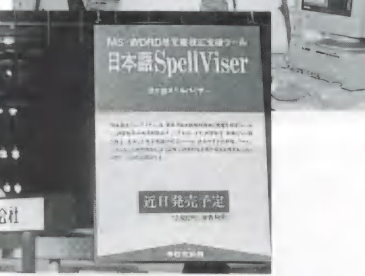
左　CONTECのノート・ラン・アダプタ。

上左 ディアイエスの「Wired for sound II」・「B–Library」・「App-roach」のブース。

上右 スターファックス対応ソフト「ナイル」・「アイリス」・「弥生」など。

左 マイクロソフトのデモ風景。

右 ロータスのビジネス表計算ソフト「1・2・3」・ビジネスワープロソフト「AMIPRO」・デスクトッププレゼンテーションソフト「FREELANCE」。

左 「一太郎」・「花子」でおなじみのジャストシステムのブース。今秋登場の「三四郎」を解説。
右 オートデスクのブース。「オートスケッチ」・「3Dスタディオ」など。

下左 ボーランドのソフト群。TorboC++など。
下右 住友金属の「ワードスター」と近日発売予定の「日本語スペルバイザー」。

左 PCエンジンのゲームソフトとイベント風影。
右 ゲームソフト・学習ソフトで遊ぶ子供達。

EL COLUMN

Business Information

"安全"と"安心"の総合サービス

～広がるホームセキュリティ～

木村 千旗

「番象」というTVコマーシャルが話題になったことがある。長嶋茂雄さんが散歩から帰ってくると，隣の家には普通の番犬がいるだけだが，長嶋家にはなんと「番象」がいるというもの。警備サービス会社のセコムが，昨年5月から1年間にわたって放映したコマーシャルだ。

「象なら安心」と思った人がいたかどうかはともかく，ここ数年，一般家庭向けの機械警備システム「ホームセキュリティ」の契約が急増している。業界最大手のセコムによれば，3年前には2万5千件だった契約件数が，今年3月には4万7千件。世をあげて不況というのに，年率23%という高い伸びを続けている。セコム，綜合警備保障，セントラル警備保障の大手3社を合わせると，契約件数は優に6万件を超える。

セコムが日本で初めてこのシステムを売り出したのは12年前。当時の契約者は特定のお金持ちか有名人に限られていたが，いまや一般家庭にも広がりはじめた。「これほど売れるようになろうとは，当時はとても想像できませんでした」とは，セコムの加藤善治郎広報室長の話だ。

住まいの安全をオンラインで管理

ホームセキュリティとは，家の中のあちこちに取りつけてあるセンサが異常をキャッチすると，警報ブザーが鳴ると同時に，警備会社のコントロールセンターに異常を通報する仕組み。要するに，

オンライン化された家庭用警報システムだ。センサの設置場所は，火事，ガス漏れ，侵入者など，監視の対象によって違う。各種のセンサが感知したデータは，室内のコントローラという装置に無線で送られ，そこから電話回線で警備会社のコントロールセンターに情報が流れる。

コントロールセンターは地域ごとの拠点となるデポ(待機所)を呼び出して，緊急要員を現場に急行させる。必要に応じて，警察，消防，ガス会社や，あらかじめ登録してある緊急連絡先にも通報を入れる。セコムの場合，コントロールセンターは各都道府県ごとに全国で47か所，デポは800か所を超え，警報を受けてから25分以内に緊急要員が現場に到着できるという。コントロールセンターもデポも24時間の待機体制をとっていることはいうまでもない。

ホームセキュリティの効用がフルに発揮されるのは，寝ているときと，留守のときだ。しかし，留守中に緊急要員が駆けつけても，家の中に入れなければ用が足りないケースが多い。そこで，デポには契約家庭の合い鍵が全部保管してある。

非常の場合とはいえ，他人の家庭に踏み込むわけだから，プライバシーの確保と悪用を防ぐ手だてが肝要。預かった鍵は袋に入れて，相手の目の前で封印し，必要が生じて袋を破ったときは，その都度，封印をやり直してもらうという。

外出時や就寝中でなくても，コントローラのスイッチを切らないかぎりセンサが常時稼働して，異常があればブザーを鳴らす。なかでも多いのは，天ぷら鍋に火が入るケースだ。天ぷらを揚げている最中に電話がかかってきたり，夕立で洗濯物の取り込みに出たりすると，鍋の中の油は平均13分で発火点に達してしまう。もうもうと立ち昇る煙をセンサがキャッチして，消防車を呼ばずにすんだことも少なくないそうだ。

押し売り撃退には暗号会話も

火事やガス漏れの警報装置は24時間フル稼働させている家庭が多いが，泣きどころは侵入者を感知するシステムだ。窓を開け閉めするたびにセンサが作動して，ブザーが鳴るのはわずらわしいので，家に人がいるときは警報装置をオフにしてある家が多い。扉や窓の開閉，廊下の人影など，用途に合わせていろいろな監視装置を取りつけても，スイッチが切ってあれば役には立たない。

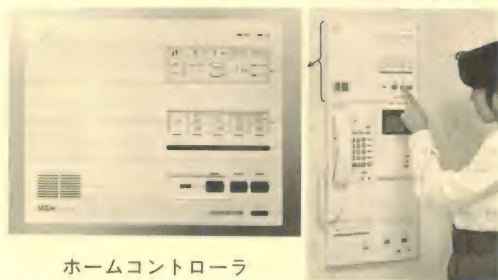
番象・番犬

今年6月，元歌手の三浦百恵さん宅に国税庁職員を名乗る男が侵入するという事件があった。玄関から追い出された男は，2階の窓ガラスを破って家の中に押し入ったが，警報装置は作動しなかった。家に人がいるので，警報装置がオフになっていたものとみられる。

センサが切ってある場合でも，コントロールセンターに急を知らせる方法はある。押し売りや強盗が家の中に入り込んだとき，ポケットに入れて持ち運びができる非常警報装置のボタンをそっと押せば，センターへの連絡と同時に外壁の赤ランプが点滅をはじめる。近所の人が気づいてくれれば理想的だが，そうでなくても，25分以内には警備会社の緊急要員が駆けつけてくる。

通報を受けたコントロールセンターが状況確認の電話を入れる場合も，侵入した相手に気づかれないように，さりげないやりとりで緊急事態を知らせる暗号の会話が取り決めてある。強盗が電話に出ても，暗号を知らないからセンターで異常を察知できるが，本人が動転して暗号を忘れてしまったらどうにもならない。押し売りや強盗が目の前にいても，あわてないことが肝心だ。

“安全”の次は“安心”を売る

住まいの安全保障と並んで，最近は病気やけがをしたときの救急サービスに対するニーズが高ま

ホームコントローラ
セコム ホームセキュリティ

セコム コントロールセンター

っている。社会の高齢化が進み，一人暮らしのお年寄りが増えているためだ。家族が一緒にいれば救急車を呼ぶこともできるが，一人暮らしで激しい発作を起こしたり，大けがをしたりすると，電話口までたどり着けないことも考えられる。

そうした事態に備えて，セコムの場合は「マイドクター」という救急呼び出しシステムを，基本サービスのオプションとして組み込んでいる。〝安全〟が売り物になるなら，〝安心〟も商売のタネになるという発想だ。

このサービスは，ネックレス・タイプのペンダントを首に掛けておき，いざというときにペンダントを握るだけで，緊急信号を発信できるという仕組み。確認の電話を入れても返事がなければ，すぐさま緊急要員が出動する一方，かかりつけのホームドクターや家族の連絡先に知らせて，必要があれば救急車の手配もする。

駆けつけたときに先方が意識不明といった事態に対処するため，緊急出動要員には人工呼吸や心臓マッサージといった緊急蘇生法の講習を義務づけている。また，風呂の中で倒れるお年寄りが多いことから，ペンダントには防水タイプも用意してあるという。

緊急時ばかりでなく，〝日常の安心〟を売るサービスも登場した。「マイケアー」という在宅健康管理のシステムだ。血圧計と心電計を内蔵した特製の椅子と，尿検査の装置を自宅や職場に備えつけておくと，警備会社の契約医に日ごとのデータがオンラインで送信されるようになっている。

毎朝5分間この椅子に座り，両手両足と左手の人差し指を5か所の金属部分に密着させておくだけで，通常の健康診断項目の70％程度をカバーできるという。検査結果に異常があれば，その場で医師から連絡があるし，正常な場合は，健康状態のリポートが毎月1回送られてくる。

こうした在宅管理のシステムは，ほかの分野にも応用がきく。例えばセコムでは，コンピュータを使う在宅学習システムの開発と取り組んでいる。学校教育へのコンピュータ導入に対応して，新しい学習システム売り込もうというわけだ。

アメリカでは全世帯の1割に普及

ホームセキュリティの契約が増えてきたといっても，日本ではまだ全世帯の0.2％。アメリカの場合は800万世帯と，全世帯の1割に普及している。アメリカでは，警備会社が地元の警察や消防と提携して，警報が流れると警官や消防隊が出動するケースが多い。サービス要員の人件費が節約できるので，料金を安く抑えられるというメリットはあるが，誤報やいたずらで出動を求めると罰金を取られる。

アメリカと比べれば治安の良い日本でも，最近は夫婦共働きで留守がちの家庭が多くなってきた。お年寄りの一人暮らしも増えている。郊外の新興住宅地に引っ越せば，隣近所とのつきあいも希薄になる。いったん装置を設置すれば，月々の料金は1万円前後だから，一般家庭が契約してもそれほどの負担にはならない。

アメリカなみの普及率はともかく，21世紀には現在の10倍の60万世帯に普及するというのが警備業界の見通し。日本でも〝安全〟や〝安心〟をカネで買う時代が近づいている。

（経済ジャーナリスト）

Electronics Essay

なんでやねん 面白IC探訪記

逆瀬川 皓一朗

趣味で電子工作をやっていると，性分なのか，パーツショップへ寄ったついでに，ついつい面白そうなICなどを物色してしまいます。

それで珍しいパーツでも見つけようものなら，「いいもの見つけた！　いずれ役に立つかもしれない」と思って買い込むんです。それも「ひょっとしたら，こんなのは二度と手に入らない」と思い予備の分まで買ってしまいますから，困ったものです。筆者のパーツボックスは(あとになってみると)いつも訳のわからないガラクタで盛りだくさんになっています。

しかし，いろいろ買い置いても，だいたいは使わないまま忘れてしまい，数年たつとクズ同然になります。それでもまあ，パーツボックスに大抵のモノがあると(思うと)，いざというときに何となく安心で，実際これで助かったこともあるので，やめられません。

こんな性分？って筆者だけかと思っていたら，ELEXのCPU.BACHさんもそうなんだそうです。似たような趣味をもっていると，性格も似てくるんですねえ。

モノ持ちはいいが…

これはと思うものを大事にとっておくのは，〝物〟だけではありません。〝情報〟もそうです。例えば新聞や雑誌の切り抜きとか，メーカーのカタログとか。そのため筆者の本箱は，ICメーカーのマニュアルやスクラップ記事で溢れています。

筆者には情報整理学なんてのはなくて，類似の情報ごとに集めて紙ファイルに挟んでおくだけですから，雑然としているんです。ですから綴じたら最後，安心して忘れてしまい，次に開いたときは黄色く変色した紙が出て来る…という始末です。ふたを開けて再び見たときは，ポンコツ同然(まるで浦島太郎？)の情報というわけです。

それでも余裕があるときには，「このICをこう使えば，こんなモノができる」などと考えを巡らすこともあります。鉄道マニアや旅行マニアが列車の時刻表を見て「旅のシミュレーション」をするのと同じように。

こんなパーツはいかが?

過日の引越しで荷物を整理していたら，案の定，むかし役に立つと思っていた情報が出てくるわ，出てくるわ。「整理することは捨てることなり」とはいうものの，全部捨てるともったいないので，面白そうなのだけをピックアップしてみました。

筆者の独断と偏見になりますが，何かのお役に立てればと思い，アナログ系のICを中心にして紹介しておきます。

【図１】USH5156の内部

●**高電圧キットパーツ**

Nチャネル／PチャネルのD-MOS FET，サイリスタ／絶縁ゲートサイリスタ(IGT)，あるいはNPNバイポーラトランジスタが独立に組み込まれたキットパーツで，500Vに耐えるところに選択価値があります。

USH5155とUSH5156(図1参照)があり(代理店はインターニックス)，エレクトロニクスの高電圧応用に重宝しそうです。

●**三角波サイン変換IC**

カスタムICですが，サイン変換マクロを使い，三角波を正弦波に変換するものです。安定な振幅・微分ひずみなし・低い波形ひずみ率・10MHzまでの応答などが特徴のようで，インターニックスが取り扱っています。

●**力率改善用コントロールIC**

マイクロリニアのML4821(代理店は日本アイシー)です。電源回路に使うICで，電源電流の高調波ひずみ率を最小にコントロールし，力率を99%以上に改善できます。

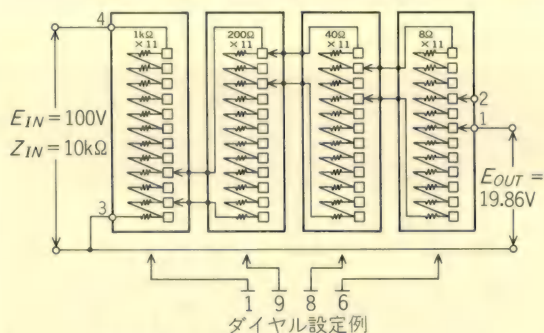
●**超低消費電力のOPアンプ**

マキシムのMAX406/MAX407で，自己消費電流が1μAの単一電源動作のOPアンプです。3Vの乾電池でも平気で使えるのが助かります。

●**超低消費電力3端子レギュレータ**

ナショセミのLM2936です。負荷がわずかなときは15μAの自己消費電流で動作し，入出力間ドロップアウト電圧もわずかな3端子レギュレータです。まあ，いまどき，そう珍しくもないかも知れませんが。

●**8サンプルホールドIC**

1つの入力に8つのサンプルホールド回路がぶら下がっているIC(アナログデバイセズのSMP-08/SMP-18)です。3ビットのディジタル入力で，どのサンプルホールド回路をドライブするかが指定できます。

【図2】QVシリーズ(電圧分割型)

●**2入力切り替えOPアンプ**

バーブラウンのOPA675/OPA676で，2系統の入力(+IN/−IN)をディジタルで切り替え選択できる広帯域OPアンプです。RFやビデオ分野で面白そうな応用ができそうです。

●**バーチャルグランド**

TI社のTLE2425で，5V単一電源の中点(2.5V)を仮想グランドにするためのICです。特に珍しいものではありませんが，5V単一電源で2電源のOPアンプを使いたいときに重宝しそうです。

●**ロータリパルスゼネレータ**

YHPのHRPG-Aシリーズ品で，超小型の光学式ロータリエンコーダです。つまみを回すと，その回転方向によって位相のずれ方が異なる2つのパルスが出てきます。その動作形式はマウスと同じようなので，マウスまがいの応用にも役立つかも知れません。

●**ディジタルポテンショメータ**

抵抗値を256段階にディジタルデータ(シリアル入力)で設定できるICです。フィルタや発振器など，抵抗値をプログラマブルにしたい応用に向いています。

ダラスセミコンダクタのDS1267S(代理店はマイクロテック)ですが，XicorのX9MME(代理店はインターニックス)も類似品のようです。

なお，10進ロータリスイッチの形状をしていて，手動で6桁までの抵抗値が設定できるポテンショメータもあります。株式会社メルソーのQV/QRシリーズ(図2参照)がそうで，これを使うとD-Aコンバータなしで可変基準電源ができそうです。

Electronics Talk

電子立国：電子秘術

清水 隆雄

●フェルマの最終定理が証明された!?

1621年刊，パシェによると3世紀から4世紀に書かれたディオファントスの「算術(Arithmetica)」のギリシャ・ラテン語対訳本の余白に，「立方を2つの立方にする場合，また，4乗を2つの4乗に分けるような，平方より大きな任意のべき(指数)をそれと同じべきの2つのものに分けることはできない。私はその驚くべき証明を見つけたがそれを書くのにこの余白は狭すぎる」とフェルマ(1601-1665年，フランス)が書いたとあるそうです。

すなわち，$X^3+Y^3=Z^3$とか，$X^4+Y^4=Z^4$....となるような整数は存在しない，と書いたというのです(それにしても，フェルマはずいぶん古い本を愛読していたようです)。

直角三角形でお馴染みのピタゴラスの定理のように，$3^2+4^2=5^2$とか$5^2+12^2=13^2$いうような関係は平方ではいくらでもあるのに，なぜか3乗以上では存在しないというのは，350年間もの間「フェルマの最終定理」と呼ばれ，数学界でも最大の難問と言われ，数に関する本にはたびたび難問として登場してきました。

これは指数n＝3(1770年，スイスのオイラー)，n＝5(1825年，フランスのルジャンドル)，n＝7(1839年，フランスのラメ)，n＝正則素数(1850年，ドイツのクンマー)，nを決めれば解はあったとしても有限個(1983年，西ドイツのファルティングス)，1988年に都立大学の宮岡洋一助教授が「完全ではないが証明できた」という騒ぎが日本でもありました。

6／26日の新聞に「アメリカのプリンストン大学のワイルズ教授がこれを証明した」という報道がありました。この難問に詳しい数学者からのコメントは(この原稿を書いている時点では，まだ)ありませんが，こうやってみると「また，あれか」と思う人も多い人騒がせな難問です。

しかも(今日ではインフレが進み金額的にはあまり価値がありませんが)，1908年には「証明した人には10万マルクの賞金を与える」とドイツのウオルスケール氏が遺言したそうです。

余談ですが，筆者はもともと数学は苦手(どちらかというと筆者の専門は英語なんです)で正確なことはわかりませんが，フランス人に言わせると「フェルマこそ微積分の発見者」と言い，ドイツ人は「ライプニッツが発見者」，イギリス人は「ニュートンが発見者」と言いはります。

日本の学校の教科書には微積分の発見者はライプニッツとニュートンは同格に扱っており「別々に微積分学を作り上げていった」とされており，コンピュータのプログラミング教科書にも「ニュートン・ライプニッツ法」という微分方程式を解く方法が書かれています。

実際にはフェルマは著書を出していないそうですが，彼はニュートンやライプニッツより10年ほど前の人です。

ライプニッツの晩年は失意のうちに世を去ったようですし，ニュートンも多忙のために婚期を逃し，一生独身で通したという話は有名ですが，コンピュータの本にはさまざまな数学者が残してくれた方法を使ったプログラミングが記されており，その理解には高度な専門知識が必要ですが実に興味をそそるものがあります(実は筆者も詳しくはわからないまま公式を利用させていただいているときが多いのです)。

今回はそんな数学的に興味をひく電子秘術をご紹介しましょう。

●電子秘術：コンピュータには競馬の予想もできる？

以下のプログラムは，かの有名な「ラグランジュの補間」と呼ばれる過去，未来や途中経過を予測するソフトウェアです。

ラグランジュ(1736-1813年，フランス)は解析力学で有名で，この頃はラプラス(1749-1827年，フランス)，フーリエ(1768-1830年，フランス)，コーシー(1789-1857年，フランス)の解析数学や微積分学の確立の時期で，約200年前にわれわれエレクトロニクス屋が日頃お世話になっている高度なソフトウェアの基礎が，この時代の先輩達が築いてくれたものであることに驚嘆させられます。

日本では「白河の清き流れに魚住まず，濁れる田沼のいまぞ恋しき」なんていう世評が流行った，松平定信のケチケチ借金棒引き令が出ていた天明の時代のプログラムです。

プログラム言語はN88BASICで書いてありますから，ほとんどのパソコンに付属のBASICで走らせることができます。

```
10 'SAVE "LAGRANGE.BAS",A
20 CLS
30 PRINT "*LAGRANGEの補間"
40 PRINT
50 INPUT "質問：曲線の通過点の数をキーインしてください……";N
60 PRINT
70 DIM X(N),Y(N):'メモリの確保
80 FOR I=1 TO N
90 PRINT "通過点";I;"の座標",
100 INPUT "(X,Y)=",X(I),Y(I)
110 NEXT I
120 '
130 PRINT:PRINT
140 PRINT "予測したい点のX座標を入力してください(0を入力すれば，終了します.)
150 *HOKAN
160 INPUT" X=",U
170 IF U=0 THEN END
180 V=0
190 FOR I=0 TO N
200 P=1
210 FOR J=1 TO N
215 IF(X(I)-X(J))=0 THE N 230
220 IF J<>I THEN P=P*(U-X(J))/(X(I)-X(J))
230 NEXT J
240 V=V+P*Y(I)
250 NEXT I
260 PRINT "の時の予測","Y=";V:PRINT
270 GOTO *HOKAN
280 END
```

このソフトは，ある関数で表すことのできるXとYのサンプルデータをいくつかキーインすると，過去・未来や途中経過を良い精度で予測するため，予測プログラムとして重宝します。

特に測定データを測定器から(GPIBで)取り込み，このソフトで計算予測する方法はあまり表だって使われていませんが，有効に利用すれば数点の測定サンプルで処理が済む場合もあって，処理時間を飛躍的に縮められる場合もあります。

余談ですが，例えば競馬の調子である周回所要タイムを過去のレースでの温度・湿度・飼い葉の量・体重などのデータと，過去のタイムとの関係をパソコンで打ち込んでいけば予測が可能かもしれません。しかし，実際にはこれらを数値化しても馬の調子を表すとは思えませんから，やっぱり競馬の予想は無理というものですが，真剣になっていろいろ予測をしようと思うようになるほど興味のあるプログラムです。

バブルの時代には株屋さんも注目したプログラムとも言われています。

Audio Visual Topics

不振のオーディオ業界をデータでみる

出原 真澄

オーディオ業界はホームオーディオが低迷し，構造不況だと言われている。今月は，その辺の事情を探ってみよう。

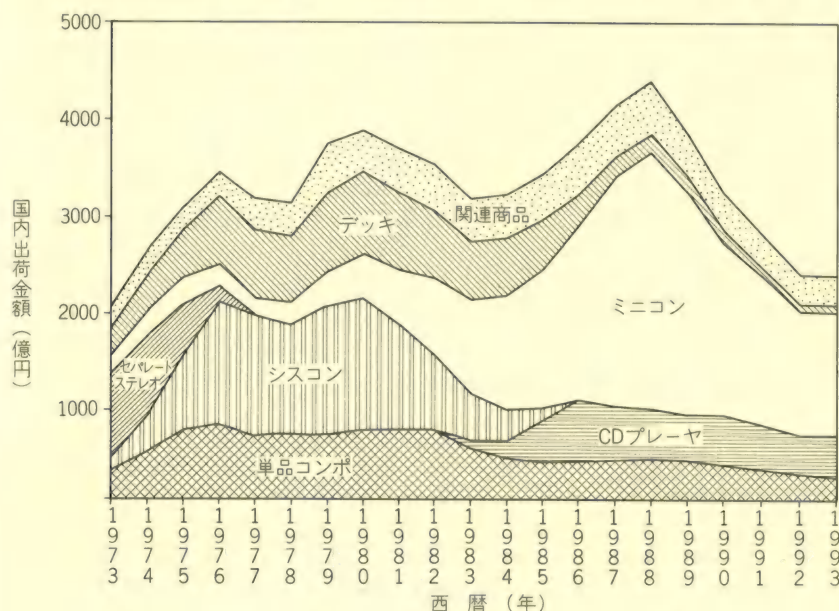
出荷高の推移

業界では用途や形態によって，単体コンポーネントやセットステレオのような置き場所を固定して使用するホームオーディオと，移動型であるラジカセ，ヘッドホンステレオなどのゼネラルオーディオ，それに車用のカーオーディオの3分野に大分類している。今回は，この中のホームオーディオの状況を見てみよう。

図1が1973年からの，ホームオーディオの販売動向である。この中で特に大きく変化しているのが一体型のステレオ(セパレート，シスコン，ミニコン等)で，その時代によって形態を変化させ売り上げ金額も大きく変動している。

単品コンポは，自分の好みに合わせてアンプやスピーカ，CDプレーヤなどを選んで組み合わせる，いわゆるマニアの世界だ。1983年から集計されているCDプレーヤは単品コンポの範ちゅうに入るもので，これを入れるとこの分野は安定している。結局CD出現後，ミニコンの販売動向がホームオーディオ全体を支配していることが，このグラフでもよく理解できよう。

それでは，キーを握る一体型のセットの動向を見てみよう。1970年代中ごろまでは,「セパレートステレオ」と称する大型のシステムが主流だったがその後，もっとデザインが良くて小型高性能なシスコン(システムコンポーネント)が主流になる。しかし，この時代は短命で，さらに小型でパーソナルなミニコンがCDの出現(1982年)とともに本格的に普及しだした。

そして約10年間，業界はミニコンとともに発展に発展を遂げた。これほどまでにミニコンが普及した背景は，当時すでにステレオの世帯普及率が50%を突破していて,「一家に1台時代」から「一

【図1】
ホームオーディオ
国内出荷高推移
(サウンドマーケットより)

人１台時代」へと突入したことにある。小・中学生が個室を占有するに及んで，ステレオのパーソナル化に拍車がかかった。

したがって小型化，高性能化が条件，ソリッドステート全盛時代となって，システムはどんどん小型化されていった。そして1982年，待望のCDが出現，これがミニコンポとドッキングして1984年からの５年間，ミニコンは史上まれにみる驚異的な勢いで成長した。ピークの1988年にはミニコン(さらに小型なミニミニコンも含む)だけで，2,635億円の出荷実績を達成した。

それがどうしたことか平成になったとたん，まるで断崖を転げ落ちるように下降が始まった。来年は持ち直すだろう，いや，再来年だといわれながらバブルが弾けたのも手伝って，昨年は全盛期の1/3近くまでに下がってしまったのである。

ラジカセやヘッドホンステレオの技術革新がホームオーディオの衰退を招いた

大衆は再生音楽にそっぽをむいてしまったのではないだろうか。音楽ソフトの出荷動向(来月のこの欄で説明する予定)を見るかぎりは，きわめて好調に販売されているから，音楽離れはしていない。

業界関係者は，「今の若者は音楽の好みが変わり，音質にこだわる音楽を聴かない」，「オーディオはネクラだから人気が無い」と，流行に敏感な若者は「レジャーやスポーツ，車に移っている」などといわれるが，いずれも説得力に欠ける。真の原因を探るのは困難だが，私は次のように考えている。

それは，CDの出現で移動型の再生機が革命的に進歩したことに関わりがあるように思う。移動型は自分とともに移動できる。そして，ヘッドホンステレオは直接ヘッドホンで音楽を聞くが，耳に直接入る音の細やかさ，左右の広がり，他人に迷惑をかけずに大音量再生ができるetc...ミニコンポの音は色あせてしまう。

さらに，歩きながら，ベッドで，電車のなかで，正にTPOなのである。機能的にもCDはランダムアクセスを可能にし，便利さは極限に達した。つまり，現代人にとって，ヘッドホンステレオで音楽のエッセンスは十分に聞ける。

そしてラジカセ。CDが出現してからの，いわゆるCDラジカセは驚異的に内容が良くなり，その音質はミニコンポをしのぐにいたった。さらに価格は年々歳々安くなる。ここに至って，高価で移動不能，そして場所を取るミニコンポの存在理由はなくなった。当然，販売実績は下降する。

このことは，CDラジカセが本格的に普及しだした時期と，ミニコンポが下降を始めたそれとが符号することからも，間違いないと思う。

空間音場の素晴らしさを体験できるシステムの誕生

先月のこの欄で「レコード演奏家論」について書いた。BGM的に音楽を流し聞きする向きには，ヘッドホンステレオやラジカセで十分だ。しかし，もっとこだわって音楽の内面に入り込み演奏家やその楽器，演奏しているホールや録音の差までも探り出そうとすると，これらでは不十分なのである。

ある程度広い空間に再生音場を創出し，そこに再現される音楽のイメージを体験するとき，深い感動で精神が高ぶる。この体験はラジカセやヘッドホンステレオではどうしても再現できない。

今われわれが積極的に実行しなければならないことは，ヘツドホンや小型のラジカセでは再現できない，次元を異にする素晴らしい音の世界を創りだすシステムの誕生であろう。その意味で単品コンポーネントの世界は，ユーザーをかなり満足させてくれる。

ラジカセをしのぎ十分に満足させ得る，一体型システムが誕生することが解決の最短距離となるが，ラジカセを追い越すことは容易ではない。むしろそれはあきらめて，単品コンポーネントとして内容が充実した，新しい時代の息吹を感じさせる，「ニューウェーブ・オーディオ」の出現が待たれるのである。それが具体的にどんなものかは私にも答えを持ち合わせていないが，今，業界全体が真剣に考えることではないだろうか。

お客さまへのアドバイス

宇稀　有理

私はマイコンシステム開発の技術コンサルタントを行っています。世間ではバブルがはじけたと騒いでいるようですが，相談は順調に増えています。

今回，その中からいくつか紹介しましょう。

X月X日

業務用プログラムが組めるくらいにBASIC言語を学びたいが，どのようにするのがよいか？（産業用機器製造業，従業員300人，資本金1億円，相談者：技術主任）

【アドバイス】

結論から先に言ってしまうと，本当にやる気があるのならBASIC言語ではなくて，COBOL言語かFORTRAN言語にしなさい。BASIC言語でも悪くはありませんが，教育機関が無いのです。BASIC言語は，その名のとおり初心者向け（BASICのBはBiginer's）の教育言語ですから，入門用としてはうってつけの言語と言えるでしょう。しかし，その反面，入門用を教える機関はいくらでもありますが，業務用を本気で教えているところは，？？？あるのですか？？？

それに比べ，COBOL言語は事務処理用，FORTRAN言語は技術計算用として，汎用計算機用の2大言語となって君臨しています。これらの言語については，入門用から業務用まで，多くの教育機関が存在しています。これらの言語は，既にパソコンでも利用できる状況となっています。

X月X日

世の中は32ビットでC言語と騒いでいますが，私はZ80CPUの能力で十分だと思っています。そこで，Z80CPUのアセンブリ言語を教えてください。（ベンチャー，従業員2人，資本金50万円，相談者：代表）

【アドバイス】

これからZ80CPUのアセンブリの勉強を始めるなんて，ナンセンスです。8ビットCPUはともかく，既に化石化しているZ80CPUのアセンブリ言語を覚えてみたところで，誰も驚きません。

『人を驚かすためにやるのではない！』と反論するかもしれませんが，マイコンとかパソコンとかは，目新しいだけで評価が替わるのが実情なのです。だれにでも知れ渡っているZ80CPUで，しかもアセンブリ言語を用いて，さんざん苦労して何かを作り上げたとしても，そしてそれが素晴らしいものであったとしても，32ビットCPUとC言語でチョロッと作ってしまったもののほうが，世間では高く評価されるのです。残念ながら，そういうものなのです。

『32ビットでやる必要性などない，8ビットの能力で十分！』と力説してみたところで，返ってあなたの技術力の低さを証明しているとしか受け止められないでしょう。製品の良い／悪いは，所詮はマイコン技術にド素人なユーザー達が決めるのです。いかに高級な技術が採用されていようが，彼らには見えないのです。また，力んで説明してみたところで，とうていわかってもらえる代物ではないのです。

となれば，技術者も技術の向上にばかりエネルギーを費やしていてはダメなのです。もっと，技術の見せ方を研究すべきです。誇張する必要はありませんし，してはいけないことなのですが，ユーザーにわからせる努力もするべきではないでし

ょうか？『どうせわかってもらえないのだから』と仲間内のマスターベーションで，憂さ晴らしや自己満足をしているのでは，前向きではないでしょう。

X月X日

外部セミナーの効果的な利用法についてアドバイスしてください。(システムハウス，従業員50人，資本金3000万円，相談者：人事課主査)

【アドバイス】

講習会やセミナーと言えば，知識を収集するために参加するというのが一般的です。ところが，私は，ほかの目的にも利用しています。

その1つ目は，セミナーのカリキュラム構成や講師陣，使用しているテキストなどを参考にするためです。実際にセミナーの主催を行ってみるとわかることですが，何のセミナーを行うにしても，一番骨の折れるのが企画段階なのです。カリキュラムとテキストさえ良いものが揃ってしまえば，誰が担当しても一応は成功するでしょう。もし，その後に講師との縁までがつなげることができたとすれば，目的は200％達成したようなものです。

2つ目は，教え方の手法・話法を学ぶことです。自分は理解していても，上手に説明できないことは誰もが経験しているでしょう。しかしこれは，天性のセンスが90％と本人の努力が10％ぐらいだと実感しています。受験予備校の人気講師がタレント並の忙しさといわれていますが，うなずける話です。

X月X日

ノイズ対策はなぜ難しいのですか？(システムハウス，従業員12人，資本金300万円，相談者：技術主任)

【アドバイス】

「ノイズ対策は難しい」とよく言われます。私も，そう思います。でも，難しいと言う前に，ちょっと考えてみたいことがあります。それは，ノイズ対策のためのツールとして，貴社では何をお持ちでしょうか？まさか，テスタとオシロなんてことは言わないでしょうね？

電気を知らない人から,「よく見えない電気がわかりますねェ！」などと言われて気がついたのですが，私たち電気屋(？)は，一般人には見えないはずの電気を，ツールを使って間接的に見ているのです。テスタで電圧や電流の大きさを見，オシロで波形を見ているのです。だからこそ，電気の状態がわかり，素人さんが不思議がる対策を行うことができるのです。

ノイズ対策が難しいというのは，ノイズが見えないからではないでしょうか？もしノイズを見ることができれば，しかるべき対処を行うことが可能でしょう。ノイズは難しいと言う人は，ノイズが見えていない人で，ノイズ屋になりきれていないのです。

では，どんなツールがあれば，ノイズを見ることができるのでしょうか？最低でも，スペアナは必要です。ノイズは，テスタで電圧を測るように，単なる大きさだけではありません。また，オシロの波形観測のように，時間軸上での変化を見るのでもありません。周波数軸上での分布状況が大切なのです。なぜなら，テレビやラジオなどの放送電波と同じ周波数の成分を多く含むノイズが発生していると，電波障害となるからです。そのため，各種の発生ノイズに対する強度規制値は，周波数帯域ごとに決められています。

電気と単位

今月は，単位のお話をしましょう。単位とは，物理的な存在，状態，現象などを定量的に表現するため，物差しとして使う尺度のことです。

昔は，センチメートル，グラム，秒を基本単位とする「CGS単位系」が使われていましたが，1948年以降，メートル，キログラム，秒，アンペアを基本単位とした「MKS単位系」に切り替えられ，その後1975年からは，国際標準化機構(ISO)の勧告にしたがって「国際単位系(SI)」に順次切り替えていくことになりました。このSI単位系は，MKS基本単位のほかに，ケルビン，モル，カンデラを加えた7種類の基本単位で成り立ち，これを「SI基本単位」と呼んでいます。ほかに「SI補助単位」として，ラジアン(平面角)，ステラジアン(立体角)が追加されていますが，これを見ると，電気に関係のある基本単位は「アンペア」だけということになります。これ以外のボルト，オーム，ファラッド，ヘンリーなどは，これらの基本単位を組み合わせたものとして表現され，「SI組立単位」と呼ばれます。例えば，「電圧：V」を基本単位で表すと，「$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$」ということ？になります。何ともややこしい話ですが，日常的には，V，Ω，F，Hなどを使うことになっています。ホッ！

アンペア(A)

もちろん電流の単位です。電流とは，物体の内部に電気の運び手である自由電子が存在し，それが継続的に移動する状態を言います。この電子は，－から＋に向けて移動しますが，電流は＋から－に流れるので，正反対です。これは，まだ電流の正体がわからなかった頃，＋から－の向きを正方向と約束してしまったからで，今なら，電子の移動と電流の向きを同じ方向に決めていたでしょう。SIの定義によれば，「真空中に１ｍの間隔で平行に置かれた，無限に小さな円形断面積を有する無限に長い２本の直線状導体のそれぞれを流れ，これらの導体の長さ１ｍごとに2×10^{-7}ニュートンの力を及ぼしあう一定の電流」が１Ａである，ということになっていますが，これではなんのことやらさっぱりわかりません。しかし実際は，この基本定義に基づき，国際度量衡委員会という組織で定めた方法によって，１Ａという電流が測定されることになっています。これをわかりやすく，電子の移動という観点から説明すると，電子はごくわずかな－電荷を持った粒子なので，これが毎秒6.25×10^{18}個の割合で移動するとき，それが１Ａの電流である，ということになります。このほうが直感的にはわかりやすいのですが，電子の数を数えることは不可能なので，測定方法としては落第ですネ。

ボルト(V)

物体に電流を流そうとして働く電気的な力が「電圧」です。そして，電圧の単位を「ボルト」と言います。電圧は，物体内部の自由電子を－側から＋側に向けて移動させる，方向性を持った力として働きます。

基本単位である「アンペア」は，電流が流れている導線間に働く力で定義されましたが，組立単位である電圧（ボルト）は，ほかの基本単位，組立単位などを使って「１Ωの抵抗を持つ導体に１Ａの電流を流す電圧が１Ｖである」，と定義されます。この場合，ＶとΩは，鶏と卵の関係で，どちらか一方が既知でなければなりません。基準電圧としては，国際度量衡委員会のジョセフソン電圧標準を用いる方法があります。しかし実際的には，標準電池，または最近使われ始めたツェナーダイオードによる基準電圧が一般的です。標準電池の例としては，1892年にウェストンが発明したカドミウム標準電池があります。硫酸カドミウム電解液の中に白金電極を２個，互いに離して設け，一方はカドミウム・アマルガムを形成して－極とし，他方は水銀減極剤に包んで＋極としたものです。この電池の電圧は，1.018620～1.018630Ｖ±30μＶです。ツェナーダイオードと順方向ダイオードを組み合わせた，半導体素子による標準電圧でも，再現性は悪いものの，標準電池以上の精度のものができているそうです。

エレクトロニクス豆知識　白土 義男

オーム(Ω)

抵抗値の単位です。抵抗とは，物体に電流を流そうとする力が働いた（電圧が加えられた）とき，その力に抗し，電流を流すまいとする，その物体固有の電気的特性のことです。これを微視的に言えば，その物体の単位体積中に存在する，自由電子の密度が高ければ抵抗は小さく，密度が低ければ抵抗が大きいと言うことになります。この「オーム」も組立単位のひとつで,「１Ｖの電圧を加えたとき１Ａの電流を流す抵抗値が１Ωである」と定義されます。かつて1908年頃には，１㎟の断面積を持つ水銀柱の，長さ１ｍの抵抗値が１Ωである，と定義されていたこともありました。イメージとしてはこのほうがわかりやすいと思います。一般的な標準抵抗器としては，巻き線型抵抗器をよく見かけます。中・低抵抗値のものはマンガニン線,高抵抗のものはNi-Cr-A1線が使われているようです。ここでちょっと「SI接頭語」の話をしましょう。これは，単位の頭に付けて，その数値の「桁」を示す補助的な記号です。例えば1200Ωを1.2kΩというように，「1000倍」を「k：キロ」という接頭語に置き換えて表現するのです。これは「SI単位系」の中で，正式に「接頭語」として示されています。身近な例では，Ｍ(メガ)，ｍ(ミリ)，ｐ(ピコ)などがあります。

ファラッド(F)

受動電気回路は，基本的にＬ(誘導)／Ｃ(容量)／Ｒ(抵抗)の３要素から成り立っています。このうち抵抗は，直流から超高周波まで，純粋に抵抗としての働きを示し，電圧と電流は同じ位相(タイミング)で変化します。「容量」は，具体的な部品としては「コンデンサ」の持つ電気的な働きのことで，２枚の電極を絶縁体を介して向かい合わせたとき，電極間に電荷を蓄積する働きが「容量」です。そして，この蓄積する能力の大きさを示す単位が「ファラッド」です。コンデンサに直流電圧を加えると，電極間がその電圧に達するまで充電電流が流れ，充電が終わると電流は流れなくなります。１ファラッドとは，１Ｖの電圧で充電したとき，１Ａの電流を１秒間流し続けることのできる量の電荷を蓄積する大きさの容量のことです。コンデンサに加えられている電圧が増加すると，その電圧に達するまで充電電流が流れ，減少すれば，電荷が放電されて電極間電圧は減少します。ですから，コンデンサに交流電圧を加えると，電圧の増加率に比例して充・放電電流が流れることになります。例えば電圧がサイン波なら，この波形の変化率，すなわちコサイン波形が電流波形となります。これを別の言葉で言えば，電圧より電流の位相が90°進む，ということです。

ヘンリー(H)

抵抗に電流を流すとエネルギーを消費(発熱)します。コンデンサは，電荷という形でエネルギーを蓄積します。そして，磁束という形でエネルギーを蓄積するのが誘導(インダクタ)です。具体的な部品としては，空芯のまま，または磁芯に導線をぐるぐる巻いた，コイルやインダクタがあります。この誘導の持つ基本的な働きは，電流の変化を妨げようとする作用です。ですから，誘導に電圧を加えると，その電圧の変化を打ち消す方向に逆電圧を発生します。直流なら，電圧を加えた瞬間と，切った瞬間に逆電圧を発生しますが，あとはコイルのわずかな抵抗で直流をショートする形になります。しかしこのとき，コイルの巻き数と電流の積に比例した量の磁束という形で，内部にエネルギーが蓄えられるのです。ヘンリーとは，この誘導の作用の大きさを表す単位で，１秒間に１Ａの割合で変化する電流を流したとき，それを妨げる方向に１Ｖの電圧を発生する大きさの誘導が１Ｈです。誘導に交流電流を流すと，その変化を妨げる方向の電圧が，流れる電流の変化率に比例した大きさで発生しますから，例えばサイン波電流が流れている場合，この電流の変化率であるコサイン波形で，かつ極性を反転した電圧が誘導に発生します。言い換えれば，誘導の端子間電圧より90°位相の遅れた電流が流れると言うことになります。

WHERE AM I ON THE EARTH?

GPS情報

山　滋

今月のトピックス

7月1日，わたしたちの生活で使用している時間に1秒が加えられました。この日は1日が24時間と1秒ありました。この1秒を閏(うるう)秒といっています。時間はセシウム原子の振動の周期を利用した協定世界時(UTC：Universal Time Coordinate)で定義されています。この原子時計の1秒と，地球の自転の変動によるずれを補正するために，この閏秒を使用しています。地球の自転が遅くなっているのがこの原因のようです。この閏秒の補正は，ほぼ毎年行われています。

GPSの世界では，GPS時が使われています。このGPS時は，閏秒による補正が行われておりません。

したがって，年々，協定世界時との差が大きくなっていきます。しかし，協定世界時とGPS時との差は，GPS受信機から放送されています。受信機では自動的にこの差をGPS時に加えているので，GPS受信機で表示されている時刻は，世界時または現地時間と同じものが表示されています。

現在ではこの差は9秒あって，GPS時のほうが協定世界時より短くなっています。

今月の話題

ノートパソコンとGPS

GPS受信機を使用した特定の応用製品を開発したい場合，GPS受信機とアンテナのほかに電源，表示装置そして操作キーが必要になります。表示装置や操作キーはパソコン，最近では，ノートパソコンがよく使われています。あるいは，最終的な製品化の場合には専用のものを開発するでしょう。

通常，GPS受信機はRS-232CやRS-422のインタフェースをもっています。適当なケーブルを用意すれば接続することができます。そして，おおよそ1ワットから3ワットの直流電源と接続コードが必要です。また，パソコン用の電源も必要です。こう書くと，簡単そうですが，実際にはなかなか大変なものです。

米国のトリンブル社は，パソコンのICカードソケットに接続できるGPS受信機を発売しています。パソコン，特にノートパソコンにはPCMCIA(Personal Computer Memory Card International Association)規格のICカードのスロットが用意されているものがあります。

このPCMCIA規格は，日本のJEIDA(日本電子工業振興協会)の規格をもとに作られたもので，世界的な規格となっています。現在使われているのは，PCMCIA Type 2またはJEIDA 4.1という規格です。この規格のカードであれば，同じ規格のカードスロットに差し込むことができます。

日本国内で発売されているノートパソコンでPCMCIA Type2規格のカードスロットを持つのは，日本IBMから最近発売されたThinkPad550BJや220があります。日本IBMではこの規格のカード製品として，2,400bpsのモデム，LAN用のイーサーネット，あるいはトークンリング・アダプタなどを発売しています。

PCMCIAカードを通じて，データの送受と電源の供給が行われます。したがって，多くの場合カードを入れるだけで使用できるようになります。**写真1**の右側の部分がPCMCIA規格のカードで，これをノートパソコンに入れます。左側の部分がGPS受信機で，この右

〈写真1〉GPSとPCMCIAインタフェース

下の部分にアンテナコネクタがあります。GPS受信機の大きさは約5×9.6×2.5cm，140gです。アンテナは直径6cmの平板型です。

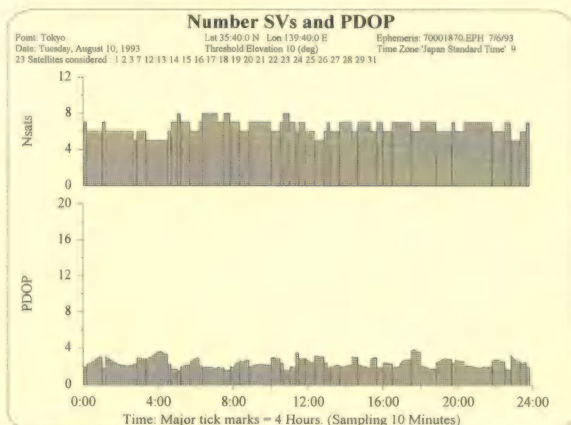
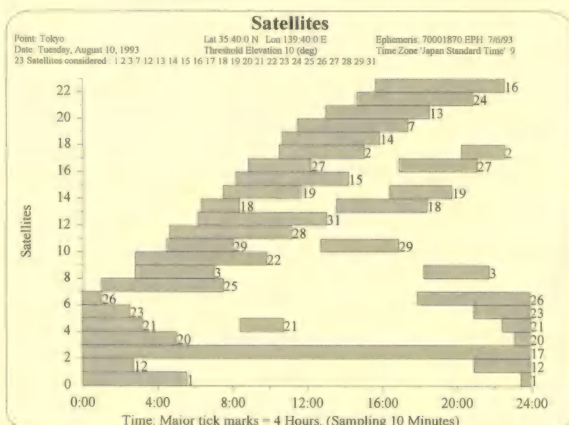
GPS受信機は6チャネルですが，8衛星までをほそくすることができます。電源はPCMCIAカードから供給され，必要な容量はアンテナの電源も含めて約2ワットです。

これから発売されるペンコンピュータや，電子手帳，パソコンなどでは，特に通信インタフェースのために，カードスロットをもつことが予想されます。各種のパーソナル情報処理システムにおいて，GPSは野外や自動車，船舶などの移動体の位置センサとして各種のアプリケーションが開発されていくことでしょう。

今月のGPS

今月は23個の衛星が運用されています。新しい衛星PRN9(SV39)は6月26日に打ち上げられました。これは24個目の衛星となるはずですが，実際には現在運用されている3個のブロックI衛星がブロックII衛星に置き換わったときが完全運用となるでしょう。今年中にはすべての衛星がブロックII衛星になるものと予想されます。

図1，2に東京におけるGPS衛星の飛来予測を示しています。

（トリンブルジャパン株式会社）

【図1】東京における受信可能な衛星数とPDOP（仰角10度以上）

【図2】東京における受信可能な衛星（仰角10度以上）

電波にも“天気予報”が……

小林 良夫

【図1】1時間の送信内容

短波の放送は，どこまでが近距離で，どこからを遠距離とするか，かなり微妙です。ラジオ日本(NHK国際放送)を東アジアで聞くのは近距離と考えてよいでしょう。直接，日本からの送信を，中東，ヨーロッパ，アフリカ，南北アメリカなどで受信するのは，完全に遠距離です。外国からの電波を日本で聞くときも，ほぼ同じわけです。

毎日，同じ時間帯に受信してみると，近距離の電波は，かなり安定しており，距離が遠くなると受信状態の変化も大きいことがわかります。

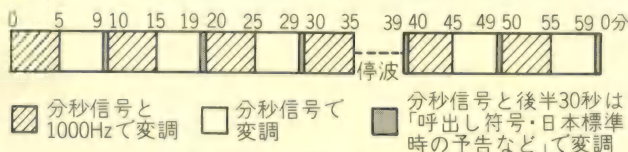
しかし，いわゆる『空中状態が悪い』ときには送信所から100km以内の『超近距離の電波』も，ほとんど受信できなくなり，受信機から出るのは雑音ばかりということさえあります。

地上の天気の場合には，天気予報や気象警報が日常の生活に必要な情報の1つになっています。

電波の伝播状態(いわゆる空中状態)についても，これとよく似た『電波警報』があります。

日本では，40年以上も前から，『標準電波』の1項目として，『電波警報』が出ています。

前号に書きましたが，標準電波JJYは茨城県にある送信所から，正確な2.5，5，8，10，15MHzの周波数で，正確な日本標準時の時刻を年中無休で放送しており，自由に受信し，利用できます。

10分ごとに，毎時9分30秒，19分30秒などからコールサインJJYと，時刻(次の0秒に対応する日本標準時の予告)に続き，5回繰り返して，モールスで電波警報を打ちます(表1参照)。

警報のNが天気予報の『晴れ』，Uは『曇り』，Wが『雨(あるいは暴風雨)』に相当します。

空中状態は，全方向の電波が揃って変化するとは限りません。方向によって違うこともあり，悪くなり方も，ときによってさまざまです。

それを，N，U，Wの3段階で表示するのですから，大づかみな傾向しかわかりません。

しかし，電波警報を聞いていれば，空中状態が悪いときに，自分の受信機や，アンテナの故障と誤解して，大騒ぎしたりしないですみます。

標準電波の周波数は，国際的に定めてあって，外国の標準電波も，同じ周波数を使っています。日本国内で受信していても，時間により，外国の標準電波などの混信が激しいことがあります。

JJYは，昔は4MHz，8MHzの2波で日本全国をカバーしていました。4MHzは30年以上前に無くなりましたが，8MHzは現在も残っています。

つまり，標準電波の8MHzは，日本独自のものですから，混信を避けてJJYを受信する目的に，海外はもちろん，日本国内でも利用しやすい場合が多いはずです。

モールス符号	文字	警報の意味
── ─	N	電波の伝播状態が安定している
─ ─ ──	U	電波の伝播状態に不安定が予想される
─ ── ──	W	電波の伝播状態に異常現象がある

【表1】電波警報の意味とモールス符号

FAXモデム専科

企画・構成　**谷沢 俊昌**

情報化時代を反映して，わたしたちの身の周りにはさまざまな情報伝達手段が乱舞しています。

パソコン通信もその1つですが，パソコン通信と同様に電話回線を利用するFAX機に，パソコンからデータを送ることはできないものか？そのような意図で開発されたのがFAXモデムであり，パソコンとFAXモデムを結び付ける役目を果たしているのが，FAX通信ソフトということになります。FAXモデムは従来パソコン通信用としてデータ送受信に使用されているモデムに，FAX送受信に必要な機能を付加したもので，日本でも会社やオフィス等の限られた範囲では利用されていました。しかし，それらのFAXモデムは汎用性に乏しく，一般のパソコンユーザーには遠い存在でした。

ところが，開発国アメリカでは既に2～3年も前から一般のパソコンユーザーの間で普及しており，FAXモデムの機種も多くFAX通信ソフトもたくさん発売されています。

アメリカの通信事情から2～3年は遅れていると言われる日本でも，今年春大手のモデムメーカーがFAX付きの14,400bpsモデムを発売したのを期に，FAX機能を搭載した14,400bpsモデムがモデムメーカー各社から発売され，14,400bps時代に，FAX通信時代に突入しました。

そこで，今回は「今が旬」のFAXモデムにスポットを当て，日本のFAXモデムの現状を探る特集を企画してみました。

1

FAXモデムの現状と近未来は

浮田 一郎

日本のパソコン事情

一般にテレビや新聞の記事では，日本は世界最先端を行くコンピュータ立国と言われている。マスコミの発表をもとにほとんどの日本人がそう信じているのである。

確かにメモリーの生産技術やカラー液晶を始めとする精密加工技術などは世界一だし，ほかにも多くのコンピュータ周辺機器の技術／生産の分野で世界の最先端を走っているのは事実である。

ところがソフトウェアの生産技術，ネットワーク通信，業務用アプリケーションなど，これからのパソコン利用に関する核となる分野において，日本の現在の状況は非常にお寒い状況と言わざるを得ない。

まず，ソフトウェアに関しては，パソコンを始めとするあらゆるコンピュータのOSのほとんどをアメリカに握られており，日本はそれを供給してもらって日本語化するのみである。また，パッケージソフトなどの分野においても世界に通用するのはゲームのみで，ビジネスソフトにおいて世界に通用するパッケージソフトなどは，皆無という惨々たる状況である。

同様にネットワーク通信，特にLANの利用状況も非常に低く，全オフィスのわずか数パーセントというのが現実である。アメリカのようにほぼ半数のパソコンがLANで接続され，ほとんどの仕事がパソコン上に行われている状況とは比較にならない。

数カ月前，幕張で「ダウンサイジングジャパン」が開催されたが，その席上でインテル会長のアンドリューグローブ氏がいみじくも日本のオフィスにおける生産性の低さを指摘していたが，LANを核としたトータルなパソコン利用が普及しない限り，この指摘のとおりの状況が続くだろう。

このような悲惨な状況となった原因としては，いろいろなものが考えられるが，おおむね，以下のような日本独自の事情が大いに影響していると思われる。

乱立するハードウェアアーキテクチャと，日本語入力方式

日本の場合，欧米などの26文字の世界と違い，日本語DOSは膨大な漢字を扱わなければならないので，初期のパソコンのハードウェアではアメリカから数年遅れてユーザーに供給されてきた。また，漢字フォントの格納場所についても，CPU性能の制約などから現在のDOS/Vのようにファイルで持てるわけではなかったので，ほとんどが漢字ROMを搭載したパソコンとなった。

これが世界のパソコンの基本アーキテクチャ(つまり，IBM-PC/AT標準機)を無視した独自のアーキテクチャの乱立につながり，日本の各パソコンメーカーのハードウェアは，すべて異なるという世界でも類をみない悲惨な状況に陥ってしまったのである。

それに加えて，パーソナルワープロという日本語タイプライタの普及がさらに混乱に拍車をかけ，現在の日本の一般オフィスでは互換性のまったくないファイルが今も毎日，累々と生み出されているという異状な状況になっている。

また，日本語における入力方式も標準的なものがなく，この事もハードウェアの乱立とともに欧米のコンピュータ利用に比較して，大きな足かせになっている。

これは日本に限った話ではなく，中国や台湾などの漢字文化圏諸国にも共通の悩みであるが，少なくとも中国，台湾ではコンピュータと言えばすべてPC/AT互換機なので，日本よりプログラム開発などの技術修得などが速く，入力方式の統一も日本より早期に標準が決まる可能性も高い。

乱立するハードウェアと日本語入力方式は，確実に日本のパソコン利用レベルを世界から大きく遅らせてしまった。

世界から孤立した最大の原因 =PC-98の普及

DOSでのビジネスソフトで世界に通用するソフトは皆無である。また，Windowsのソフトに至っては，開発できるプログラマさえ数えるほどである。

いまさら誰を責めるわけにもいかないが，これほどまでに日本のビジネスソフトウェアが世界のレベルから大きく遅れた最大の原因は，PC-98の普及にある。

これまで日本のソフトハウスのほとんどは，開発のターゲットをPC-98に置いており(確かに当然であろうが)，日本語の使えない世界標準アーキテクチャであるIBM-PC/ATを無視し続けてきた。日本IBMのだらしなさも一因である。

すべては移植，移植に明け暮れ，同じintelのCPUを使いながらも海外製ソフトウェア移植とサポートにエネルギーのほとんどを吸収され続けたのである。

これは日本語Windows3.1Jが登場した今も続いており，DOSよりは楽になったとはいえ，ハードウェア固有の問題の悩みはまだ続いている。

DOS/Vの登場は，日本人がそのままの形でPC/AT互換機を利用できる初の環境であり，これでやっと世界の仲間入りを果たしたことになるが，あまりにも遅きに失した登場であった。とはいえ，今後の10年間を考えるとき，今はまだ小さなDOS/Vの芽だが，根は世界標準の大きな根に直系なので，今後は大きな森林に育つ可能性は十分にある。

世界に通用するレベルの開発者を，日本だけの狭い世界で衰弱させてはならないのだ。

さて，冒頭から日本のパソコン事情と題して，PC-98ユーザーには失礼なことを言ってしまったが，今後はぜひ英語などの修得を心がけ，広く世界を見ることをお薦めする次第である。

この特集である「FAXモデム専科」で登場するFAXモデムひとつとっても，今後は日本の製品は弱体化し，世界レベルの安価で優れたハードウェア／ソフトウェアが続々と登場することになる。

そのとき，PC-98シリーズでは使えない製品がかなりの比率になるだろう。

FAXモデム購入を期に，広く世界のコンピュータ市場に目を向けてみよう。地球には日本人だけが住んでいるのではないことをしっかり認識できるはずである。

FAXモデムのメリット

モデムとFAX機能が合体した FAXモデムのメリット

FAXモデムとは従来，別々の機能として存在したパソコンFAXとパソコン通信の機能を一つのハードウェアで実現したものである。

共にパソコンの拡張スロット，もしくはRS-232CというI/Oポートを使い，共に相手のデバイスと通信をするという意味では，ラジカセの場合の合体の必然性と同じである。違うのはパソコン通信が全二重という同時双方向通信方式なのに対し，FAXのそれは半二重という通信方式であることの違いくらいである。

それならば拡張スロットやRS-232Cを共有化して使おうというのは極めて自然発生的な考え方であり，これがFAXモデムの基本的発想となっている。パソコン通信もFAXもそれ自体，同じモデムであり，拡張スロットやRS-232Cを共有して1台にしたメリットは非常に大きい。

G3FAX機は1課に1台, FAXモデムは1人に1台

FAXモデムの価格下落は恐ろしいほどのスピードで現在も下降中である。これはFAXモデムが個人のパソコンで利用するため，複数の人間で利用するG3FAX機の台数より多く普及するからである。

FAX機能を搭載したFAXモデムチップセットは，急激に低廉化している。EIA/TIAという機関でCLASS1，CLASS2のFAX機能の標準化が決まったおかげで，モデムチップを生産している各メーカ

ーとも通常のモデムチップと変わらない値段で，FAXモデム用チップセットを供給できるようになったのである。

今後，われわれは14,400bps/14,400bpsのFAXモデムを3万円前後で購入できるようになるはずである。

目に見える形でオフィスのペーパーレス化が可能

現在のオフィスでは，パソコンやワープロでデータを作成することは当たり前になりつつあるが，作成したデータをFAXで送る場合，ほとんどがプリンタで紙に印刷してからFAXで送っている。

これでは時間の無駄でもあり，FAXが終わってから紙を捨てることにもなるので，資源の無駄使いでもある。

FAXモデムを使えば紙に印刷せず，そのまま相手のFAXに送れるので作業時間も短く，紙も無駄にならない。また，個人で使用できるのでFAXの順番待ちからも解放される。

鮮明なFAX出力が可能

普通のFAXでは，紙に書かれた文字や図形などを光学的に読み取り，それを電気信号に変えて電話回線に送り出すため，読み取り時にニジミやカスレが発生し，地図などの細かい文字などはつぶれて読めない場合が多い。

しかし，パソコンFAXでは，原稿は最初からパソコンのデータとして光学的読み取りを必要とせず，ダイレクトに画像を送ることができるので，非常に鮮明なFAX出力が可能である。特に，WindowsやMacintoshなどのGUI環境でFAXモデムを使えば表現力も倍加し，極めて鮮明かつ説得力あるFAXを送れる。

ほとんどのアプリケーションからダイレクトにFAXできるメリット

DOS上でパソコンからFAXを送る場合，必ずFAXソフトを起動してファイルを指定する必要があったが，Windowsなどでは印刷機能を装備したアプリケーションであればワープロ，表計算ソフト，グラフィックソフトなど，すべてのアプリケーションからダイレクトFAX送信が可能である。

操作もはるかに簡単であり，画面イメージを相手のFAXにそのまま送れるメリットも非常に大きい。

マルチ(疑似)動作で，バックグラウンドFAXが可能

さらに，これもDOSの制限であるが，パソコンからFAXを送っている間は，ほかの仕事は何もできなかった。送信している間は黙って待つしかなかったのである。

しかし，Windowsなどの疑似マルチタスク環境下では，マシン環境が許す限り，複数のタスクを動作させることが可能となり，バックグラウンドでFAXを送信しながら同時にほかの作業をいくらでもこなすことが可能になった。

実際に使ってみると，非常に便利であり，仕事の効率化という点でもDOS環境での作業とは比較にならないほどの差がある。

FAXモデムのデメリット

ダイレクトに紙をハンドリングできないもどかしさ

このようにパソコンを使う場合に非常に便利なFAXモデムだが，それではデメリットはないかというと，確かにある。

まず，ダイレクトに紙を扱えないために紙の情報のハンドリングが苦手であるという点である。

この問題の克服はFAXモデムが持っていない機能，つまりスキャナとの併用で実現できるが，まだまだ操作や実装などの面であまり

一般的ではないようだ。

しかし，いずれLANが普及すればカラースキャナなどを多くの人が共有し，今のFAXではできないカラーFAXなどが実現できるので，今後に期待したい。

操作の難しさ

紙をダイレクトにハンドリングできないもどかしさと共に，FAXモデムのデメリットとしては，パソコン操作の知識が必要だということだろう。

まず，キーボードを操作できなければダメだし，ファイルの概念を知っていなければFAXを送ることはできない。

それでもいずれLANが普及してくれば，オフィスワーカーの世界はパソコンなしでは仕事などできなくなるであろうから，いずれは解決するだろう。

FAXモデム時代は到来するか？

これは必ず到来する。すでにアメリカでは高速モデムに限っては，すでにFAXモデムが従来のモデムより多くなっているし，価格もほぼ同じくらいになってきている。

もし，片方にテレビが5万円で売られており，その横に同じ5万円でビデオ付きのテレビデオが売られていたら，普通の人はどちらを購入するだろうか？それと同じである。

なお，今後FAXモデムが主流となると思われる原因をいくつかあげてみたが，すでに日本でもNIFTY-Serve「モデム&移動体通信フォーラム」(FMODEM)などの書き込みを見ても以下のような傾向が顕著になりつつある。

WindowsやMacintoshなど，GUI環境への移行

Windowsの爆発的普及に伴い，Windowsアプリケーションからダイレクトにfaxが送れるFAXモデムは，ビジネス利用を中心に世界中で非常に注目を集めている。

DOSが主流の時代には，これほどFAXモデムが普及するとは考えられなかった。DOS上ではFAXモデム側のハードウェア，もしくはソフトウェア側でフォントを積む必要があり，文字数，画数ともに膨大な日本語や中国語などでは，ハードウェアにフォントを搭載することが困難だったため，ほとんどがファイルの形で漢字フォントを1種類，FAXソフトから呼んでいたのである。

ところが，WindowsやMacintoshのように最初からスケーラブルなアウトラインフォントを搭載した環境下では，FAXソフトは自前でフォントを持つ必要がなく，プリンタドライバのような構造だけで非常に表現力のあるFAX出力が可能になった。このため現在，FAXモデムの市場は非常な勢いで拡大している。

FAXモデムの標準APIの普及

数年前のパソコンFAXでは，ハードウェアとソフトウェアのAPIに標準的規格が存在しなかったため，ほとんどの製品が個々のハードウェアに依存した形で専用のFAXソフトウェアを添付していた。

これらの製品にはソフトウェア相互の互換性がなかったため，市場規模も非常に小さく，マニアだけの世界の様相を呈していた。

この状況を打破したのが，EIA/TIAで決められたCLASS1，CLASS2と呼ばれるFAXモデムの標準規格であり，データモデムのATコマンドでFAX送信が可能なように拡張され，ハードウェアとFAXソフトとの間のAPIが規定されたので，異なったFAXソフトで異なったFAXモデムが動作するようになったのである。

FAXサーバーの普及

LANが普及していない日本でも，パソコンを使って仕事をする以上，これからは必ず普及するだろう。そこでは，多くの社員が情報や周辺機器を共有し，より高い生産性を追求していくことだろう。

FAXサーバーは共有プリンタと同様の考え方で，多くの人間でFAXモデムを共有するシステムである。

ユーザーは自分のパソコンから印刷するのと同じ方法で，世界中のFAXマシンへFAXを送ることができる。しかも，自分のパソコンにFAXモデムを装着する必要はない。プリンタと同様に共有しているからである。

すでにMicrosoft Mail3.0やLotus cc:Mailなどでは，電子メールとFAXサーバーのドッキングシステムがあり，非常に便利な使い方ができるようになっている。

2

FAXモデムの概要と動作原理

北久保 俊

モデムのFAX化

7年前300ボーのデータモデムが登場してから1,200bpsへ，そして2,400bpsを経て14.4Kbpsへと，もっぱら通信速度を上げるのが専売特許のようだったデータモデムに，変革の兆しが見えたのは3年前のことだった。

きっかけを作ったのは「FAXアダプタ」と呼ばれるもので，データモデムから派生した品種である。これはパソコンでFAXを送受信できるようにするためのモデム装置で，それなりの需要があった。これを使えば普通のFAX装置では送ることのできない密度の濃い原稿がパソコンからダイレクトに送受信できる。

いくつかのリサーチ会社の調査や雑誌のユーザーアンケートによると，これらFAXアダプタの購入理由は，

①プリンタで紙に印刷してから，それをFAXの原稿台にのせて送る2度手間は面倒。FAXアダプタを使えば直接送れるので便利。
②密度の濃いきれいな印字でFAXが送れる。

という2点に集約される。そのほか，「イメージスキャナの代わりに使える」など，メリットとしてあげる項目は多いが，特殊な例を除き実際には使われていない。

このFAXアダプタは，モデム装置には違いないが，パソコン通信などデータ通信には利用できない違和感があった。この「違和感」が後のFAXモデム登場の下地を作ることになる。

FAXアダプタのメーカーもOEMを含め最盛期には5社ほどになったが，シェラやロックウェルなどモデムチップ(デバイス)提供者がその市場性に着目し，一般のデータモデムチップにFAX通信機能を追加したことで状況は一変した。

PC用のメモリーボードなどを提供する中小の周辺機器メーカーがさらなるベンチャービジネスとしてFAXアダプタを手がけたこともあり，投資体力に乏しく迎撃を模索する動きもなく(FAX専用である)，FAXアダプタはこの時点で事実上消滅した。

FAXモデム登場

いわゆるモデム(パソコン通信で利用するモデム装置)は，オムロンやアイワ，マイクロコアなど中規模企業が開発したビジネス分野だった。少なくともデータモデムである限りは，今後もその地図は塗り変わらないだろう。

しかし初めてのFAX，データ兼用モデム(FAXモデム)は，メガソフトやエプソンが発売したが，その後の機種では家電大手の松下がFAXモデムをもってモデム市場に参入し，ソニーも独自のホスト機能を持つメモリー機能内蔵モデムを相次いで発売した。

現在はアイワ，オムロン，マイクロコアとも価格戦略に打って出て14.4Kbpsの高速FAXモデムを安価で販売している。それに海外メーカーの参入も予想され，データ交換だけのFAXモデムならメーカーを問わず，安いものが買い得という見方が支配的になってきた。

今後は「価格指向のFAXモデム」と(メモリー搭載など)「インテリジェントなモデム」とに2局分化していくだろう。

FAXモデムの種類

発売しているメーカーの数の割にはFAXモデムの種類は少ない。もし，このような特集で全モデルの機能一覧表を作ったとすれば，どの機種も同じ機能に丸が付きユーザーはどれを選べばよいかわからなくなる。せいぜいデザインが違うくらいで，モデムデバイスメーカーの推奨回路をそのまま作る例が多く，性能もどれも同じと思って大違はない。

やはりジャンル分けをしないと実態はつかめないので，機能による分類をしてみることにする。やや大胆だが，FAXモデムは以下の3種類に分けられる。

①SENDFAXモデム
②CLASS 1， 2対応FAXモデム
③メモリーFAXモデム

①SENDFAXモデム

FAX機能は送信専用で初期の機種に多い。代表的にはシェラのモデムチップファミリーで機種数も多く，WinFaxやBitFaxなどのFAXソフトもこのSENDFAXはサポートしている。

②CLASS 1， 2対応FAXモデム

日本の電子機会工業会(EIAJ)に相当する米国の団体EIAが定めた送受信FAXモデムに関する規格で一般にCLASS 1，2という。利用者にとって「CLASS」がどのような意味を持つかの観点で説明すると，はっきり言ってCLASS 1も2もまったく差はない。

FAX送受信の基本構造は，CCITT(ジュネーブにある世界の標準を決める機関)の規格T30で定める接続するまでの処理(俗にトレーニングと言い，FAXの発着信時にピーヒョロヒョロ～という音が聞こえる部分)と，T4で定める画データのMH変換処理からなっている。

この中でT30の処理を「パソコン側で実行する」のか「モデム側で実行する」のかの違いがCLASS 1と2の違いになる。つまり，CLASS 1とはT30の仕事である独自モードが有るか無いかの識別，相手FAXの許容速度，回線状態を考慮した適正通信速度の決定，ファイン・ノーマルの識別などをパソコン側のソフトでこなしている。

これに対し，CLASS 2はその仕事(T30)をモデム側でこなすことができる。この仕事は面倒だが，決まりきった内容なのでいったん作ってしまえばモデム側ソフトに組み込める。モデムデバイス(モデムチップ部品)メーカーは，初期にはモデム本来の機能である変復調部分しか提供しなかった。そのため，通信ソフト会社は苦労してT30をパソコン側で実行できるようにしたが，現在は既にそれをモデム側に組み込みCLASS 2のモデムチップを出荷している。

このように接続までの部分は決まりきった仕事なので，パソコン側で実行しようとモデム側に任せようと利用者にとっては何の意味もない。機能が増えるとか使いやすくなるといった利用環境の向上とはまったく無縁のものである。あえて言えば，CLASS 1より2のほうが信頼性が高くなったことだろう。事実，バックグラウンド利用でCLASS 1に設定すると，ハングアップするなど問題のあるソフトもあるようだ。

このCLASS概念は，「親(パソコン)離れの程度をいう」もので，次のような**表1**にまとめることができるだろう。

③メモリーFAXモデム

FAXモデム先進国の米国では，いくつかの混乱を経ながらメモリーFAXの方向に進もうとしている。FAXモデム登場の初期には，「FAXモデムを買えばFAXは不要」という誤解があった。

しかし，実際に普通のFAX装置の代用に使おうとすると，いつ送

【表1】
機能とCLASS区分概念

機能 ＼ CLASS / 親離れ度	CLASS1	CLASS2	CLASS3?
	小	中	大
画像データのHM変換（T4）	X	X	X
FAX接続までのﾄﾚｰﾆﾝｸﾞ（T30）	X	○	○
時間指定／同報送信／エラー管理 マルチタスク処理	X	X	○

られてくるかわからないFAXのために，パソコンを常時待機させていなければならないとか，所詮，紙に打ち出さないと用をなさないということが理解されはじめ，FAXモデムは(特殊な用途を除いては)，送信専用という認識が定着しつつある。

現在，米国ではFAXモデムの最も有効な利用法も理解され，電子メール，普通のFAX装置，そしてFAXモデムが合理的に使い分けられている。その中でも特徴的なのがFAXメールで，多数の人に安いコストでメッセージを届けるのに最も適しているのが認められ，データベース型のFAXメールソフトも登場しつつある。

FAXモデムをよく利用する人にはCLASS 1，2のモデムでは不都合がおきてきている。多くの相手にFAXを送っている間，パソコンが事実上使えなくなることが問題視されている。また，国際FAXなど時間指定送信する場合もパソコンの電源が落とせなかったり，現在送信中の場合は別の場所に送るFAX登録ができないのがあわせて問題として認識されだした。

このような状況下でインテルがIBM-PCの拡張スロットに挿入するメモリーFAXモデムを登場させた。いわばFAXアクセラレータなのかもしれないが，パソコンの電源を落とせない点やトータル的なネットウェアとして捉えると，高価なメモリーをただこれだけのために使うのはもったいない気もする。

話しが少し横道にそれたが，米国の状況をよく観察すると日本も同じ道をたどりそうだ。日本ではパソコンの電源を常時入れっぱなしにできる環境はなく，またモデム単独でFAX送信や受信をこなしてくれる機種の登場が期待されている。

しかし，実は日本にはメモリーを持ったFAXモデムが2つのメーカーから発売されており，この点では米国の3歩後を追いかけていると言われ続けたモデム業界も，やっと追いつき追い越そうとしているところが出てきている。HAL研究所とソニーの製品がそれだが，モデムに内蔵されたメモリーにFAXデータを書き込み，それを指定された複数の宛先に同報送信することができる。

これからのことを考えると，メモリー送信がもたらすメリットは大きく，FAXモデムでCLASS 1や2の違いが利用者側には何のメリットをもたらさないのと異なり，5ページ程度のFAXを10か所に送ると1時間以上パソコンが使えなくなってしまうこともなくなり，また遠方へのFAX送信なら時間指定することにより通信費も安くできる。

日本にはこうしたイテリジェントなFAXモデムがあるので将来は楽しみである。ややハード先行の感はあるがこれらのメモリーFAXモデムをサポートした良いソフトが数多く出ることを期待したい。

FAXモデムの動作

前述のようにFAXモデムが処理する内容は大きく分けて2つある。一つがT30の処理でFAXを送りたい相手を呼び出した後，300ボーの速度でお互いに名乗り合い相互のFAXの素性を調べる。

次にお互いが認識できる最高速度から試験送信を開始し回線の品質を確認する。もし品質が悪くデータを取りこぼすようなら少しずつ速度を下げながら通信に障害の

①ＩＤ識別	SF/SMDFAX等	nバイト(主に製品ID)
②リゾリューション	0(30h)/1(31h)	1バイト(ﾉｰﾏﾙ:0/ﾌｧｲﾝ:1)
③最大記録長	0(30h)/1(31h)/2(32h)	1〃(A4:0/1:無制限/2:B4)
④ＦＡＸデータ	1ページ目データ	EOL/LINE DATA/EOL・・・・・・
⑤改ページデータ	ＲＴＣ	0hX4/EOL/EOL/EOL/0hX4
⑥ＦＡＸデータ	nページ目データ	EOL/LINE DATA/EOL・・・・・・
⑤改ページデータ	ＲＴＣ	0hX4/EOL/EOL/EOL/0hX4
⑦ＦＡＸデータ	最終ページデータ	EOL/LINE DATA/EOL・・・・・・
⑤改ページデータ	ＲＴＣ	0hX4/EOL/EOL/EOL/0hX4

【表2】
FAXファイル
フォーマット

無い最高速度を判断する。これを俗にトレーニングと言い，その結果により速度を落としていくことをフォールバックすると言う。

CLASS 1ではNSF, DISなどいちいちステップごとにタイミングを取りながらパソコン側で画データを送ってよいところまでお膳立てしなければならない。「CLASS 1，2対応モデム」の項目でも触れたが，CLASS 1ではこのような重い処理をパソコン側でやらなければならないので，バックグラウンド操作をさせるFAXソフトでは負荷の大きい中での作業となり技術的にも難しく，またバグの要因にもなってくる。Windowsで動くFAXソフトでも機種互換性は無いと思ったほうがよく，IBM互換機以外では海外ソフトの移植版は使えない。

こうした面ではCLASS 2のほうが楽で，ATコマンドの拡張(AT+Fの実行)でモデムのモードをFAXに切り替えATDでダイヤルし，FAXのCONNECTが戻ったらHM加工されたFAXデータをモデムにアップロードすれば簡単にFAXを送ることができる。

FAXファイルのフォーマット

FAXモデムではパソコンを操作しプリンタで打ち出すイメージそのまま相手のFAXに出力するような使い方になる。いわゆるプリンタエミュレーションでA4の場合横1ラインに1728ドット，縦1㎜にファインで7.7本の解像度を持つ200dpiのプリンタとして扱う。

この範囲に画像データ展開し，1行ごとにHM(モデファイド・ホフマン)圧縮していく。この圧縮形式は次に詳しく説明するが，FAXファイルの頭にはいくつかの制御データも埋め込まれている。

FAXソフトで画データをFAXファイルに変換し，その中身を調べてみると，一般には**表2**のようになっている。

①ID識別

ファイルの先頭nバイトにアスキーでFAXソフトのIDなどが書き込まれている。ほかのソフトで変換したFAXファイルを送ることができないのは，そのIDが異なるためである。STARFAXの場合は，SFと記述されている。

②リゾルーション

nバイト目のコードがリゾリューション(解像度)を規定する。アスキーの1がファインでゼロはノーマルを選択する。パソコンFAXではほとんどの場合，1のファインを使う。

③最大記録長

縦方向の記録長で，アスキーのゼロがA4，1が無制限，2がB4を示す。通常，ほとんどの場合A4を指定する。B4は正確なB4サイズを意味せず，単に縦方向がB4と同じ長さになる。

④FAXデータ

A4サイズのFAXデータは，1ライン(横方向)が1728ドットで，縦にノーマルモードでミリ3.75本，ファインで7.7本の解像度を持つ。

FAXデータには1ラインの先頭にEOL(ライン終端符号)を付加し，次にFAXデータ，そしてFAXデータの1ラインの終わりにはEOLを付加する。つまり，1ラインのデータの前後はEOLがなければならない(**表3**)。

FAXデータはビット単位だが，パソコンで扱う場合はバイト単位が便利なのでバイト単位で扱う。そのため，EOLやRTCなどの符号はバイト単位の切れ目から始まら

【表3】1行のデータ

1ライン	EOL	DATA	
2ライン	EOL	DATA	
nライン	EOL	DATA	
最終ライン	EOL	DATA	RTC

バイト単位	0h	28h	9Bh	15h	←データは下位ビットから使用
ビット分解	00000000	00010100	11011001	10101000	
符号　分解	00000000	0001			EOL(ライン終端符号)
		0100	11011		白1728(MAKEUP符号)
			001	10101	白0　(TERMINATE符号)
				000	0で埋めた余りビット

【表4】
1ライン白の場合

なければならない。また，バイト単位なので余ったビットはゼロで埋めなければならない。

次に例として，1ライン白の場合を示す(**表4**)。

⑤改ページデータ

ライン終端符号であるEOLを2個以上連続で送出した場合は，改ページのための制御復帰信号(RTC)と見なす。ただし，RTCの前後には0hが連続4バイト以上なければならない場合もある。

FAXソフトの種類と現状

FAXソフトは使用するPCにより，DOS版，Windows版，MAC版と事実上3つに大別できる。

①DOS版

現状入手できるのはPC-98用がほとんどである。クラス1機種汎用のSTARFAXやソニーのSMD専用のSMD-TALKがある。DOS版は，実績もあり動作が安定している。

②Windows版

Windows版ならIBM-PCでもPC-98でもあるいは他機種でも動作するはずと思うのだが残念ながらそうではない。「MS-DOSが初めて登場したときに，これで16ビットパソコンは，メーカー問わず同一のソフトを動かせる」という誤解があったがWindowsでも同じことを繰り返そうとしている。

MS-DOSで外部記録フォーマット(ディスク)の互換，テキストまでの画面互換が達成された。Windowsではグラフィックスとプリンタの互換が達成されたと思ってよい。つまり，それ以外は問題のある部分もある。特にRS-232Cに対する対応はきわめてお粗末で，MS-DOSと同じく「Windows for PC 98」とか「Windows for IBM」というようになるだろう。

WinFax，BitFax，SUPER FAXなどプリンタドライバ形式の海外製(ほとんど米国)FAXソフトがよく知られているが，IBMでは動くが，PC-98用でないと完全には動かない。多数の宛先に送る同報送信や時間指定送信など，バックグラウンド動作させる場合は表のアプリケーションをハングアップさせる場合もある。

③MAC版

OSが7になってからコミュニケーション・ツール・キットの提供もあり，日本人提供のソフトも続々と登場するようになるだろう。現在あるのはFAX-STAFFなどプリンタドライバ型のFAXソフトで，CLASS1，2をサポートしている。

10月にはメモリーモデムをサポートしたFAXソフトの登場も予想され，そうなれば同報送信でもPCの速度を落とすこともなく，また時間指定送信でもPCの電源を切って帰れる。

パソコン通信からスタートしたデータモデムは，FAXの変復調器をデータ通信用変復調器と一体化させCLASS 1規格のFAXモデムを登場させた。やがてその動きはT30の処理をパソコンから解放し，MH圧縮した画データを扱うだけでFAXの送受信ができるCLASS 2になった。次のステージでは，パソコンが完全に解放されるメモリーFAXモデムが業界の主流になる。しかしハード先行でソフトの充実がいま最も望まれるところである。

これらの動きは，一種のマルチメディア化で将来はさらに別種のインタフェースも取り込もうとしている。その最有力候補は音声録音再生機能で，電子メール，FAXメール，ボイスメールへと発展していくことだろう。

FAXモデム・メーカーのコンセプトとは

①アイワ

上田 順祚

今年の春からアイワはファックスの送信，受信機能が付いたFAXモデムを4機種発売しました。どの機種もたいへん評判がよく多くの方々にご愛用いただいており，これからますます普及する方向にあります。このファックス機能について簡単に解説をして内容を知っていただき，アイワのFAXモデムの便利な機能をより多くの皆さまに広くご利用いただきたいと思います。

電話回線を使用した通信には，モデムを使用したデータ通信のほかに，たいへんポピュラーであるファックス通信があります。ファックスは，紙に書かれた文字や画像を簡単に送ることができる大変便利な通信手段です。ファクシミリマシンどうしの通信として全世界的に普及しており，今では少なく見積っても2,000万台以上が設置されていて，オフィスやホテルなどでは100%に近い普及率で，これから家庭用の普及が見込まれます。

FAXモデムの誕生

パソコンからの情報を既存のFAXマシンに直接送り出し，また受け取るためにFAXモデムが誕生しました。

FAXマシンの中にもモデムがあります。データ通信用のモデムにファックス機能を追加することは，構成がたいへん似通っていることから，ほとんどコストが掛からないで機能追加ができるという特徴があります。

FAXモデムの通信は，図1のようにデータ通信はもちろん，一般のFAXマシンと直接通信ができます。

パソコンで作ったワープロのデータや図画をプリンタを使って紙に打ち出す必要がなく，直接FAXモデムを通して相手のファクシミリマシンに送り付けることができます。

同様に，先方からのファックスを受け取ることも可能でディスプレーで見たり，プリンタに打ち出すこともできるのです。

携帯用の小型のFAXモデムを持って行けば，旅行先からノートブックパソコンで作ったワープロの文章をプリンタ無しで直接本社などに送り付けることができます。

たいへん身軽に旅行ができますし，プリンタに打ち出した以上にきれいな文字を相手のファックスに送ることができるので，その効果も十分です。

ファックスの長所

ファックスの良いところは，FAXモデム側からみると，

①リモートプリンタとして使える（全世界のFAX2,000万台と接続が可能）

②時間を選ばずいつも待機状態になっている

③機種を選ばないのでどこのファックスとも送受できる

④鮮明な文字が送れる（ディジタル情報がドットで送れるため）

このような点がFAXモデム側から見た通信相手としての一般ファックスの大きな利点です。

EIA578規格

1991年ごろからUSAで急速にファックス付きのモデムが普及し始めました。これはFAXモデムをアメリカのEIAで取り上げ標準化の方向を目指し，EIA-578として統一規格を制定したからです。

統一化されることにより急速に市場が大きくなったのです。アイワのFAXモデムも，このEIA578に準拠しています。

EIA578は，次のことが基本になっております。

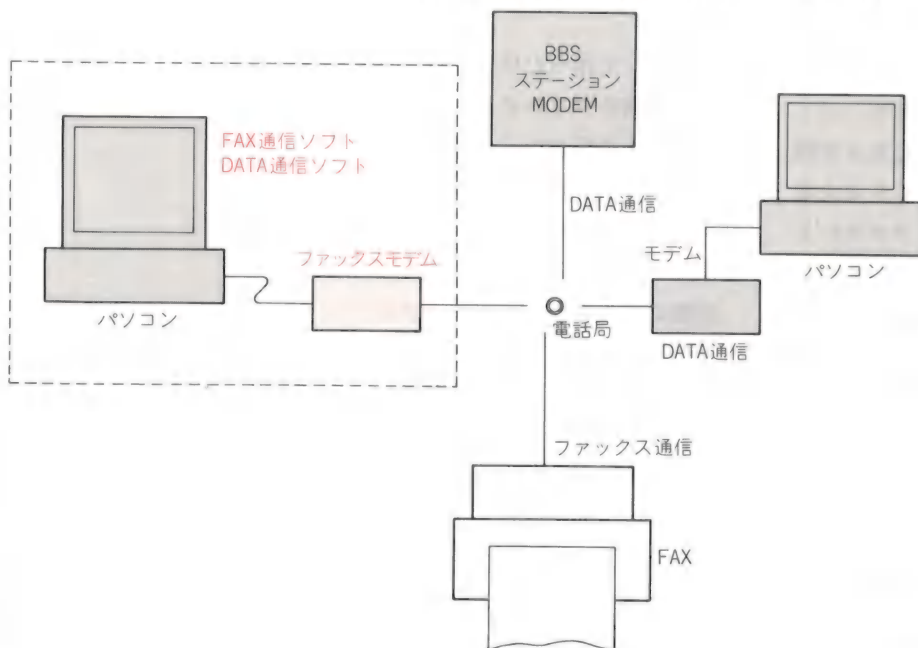
*RS-232Cシリアルポートでファックスデータを送る。

*FAXモデムの制御をATコマンドの拡張で定義。

*FAXモデムとファックス通信ソフトウェアで送受を行う。

たいへん当たり前のことのようですが，統一して各社が同じ規定を守ることはたいへん重要なことなのです。

NECパソコンへの対応

NECのある機種では，FAXモデムが使用できないという問題があ

【図1】
FAXモデムの
通信相手

りました。EIA578規格で，パソコンとのインタフェースは，RS-232Cと規定されております。ファックス通信を行うときRS-232Cのスピードは19,200bpsに設定することが必要です。

NECのパソコンは，クロック周波数が8MHz系と10MHz系の２種類があります。8MHz系では，分周比の関係からこの19,200bpsにセットすることができないので，このままでは通信が不可能です。

対応策として，アイワのモデムでは8MHz系のパソコンの場合，分周比から得られる一番近い値の20,800bpsを使用する方法で解決しています。

①通信ソフトウェアで，パソコンのクロック周波数を調査
②8MHz系の場合モデムに状況を知らせる
③RS-232Cのスピードを20,800bpsに設定
④ファックス通信開始

上記のような手順にて8MHz，10MHz変わりなく通信ができるようになっています。

美しいファックス画像

ファックスの分解能は，ノーマルモードで3.85本／１mmであり，ファインモードでさらに２倍の値を持っています。FAXモデムでパソコンのワープロなどの情報を送る場合，このファインピッチを有効に利用してたいへん美しい画像を相手に届けることができます。

これは，ファックスのドットに合わせてパソコン情報を割り付け送り出しているからです。

通常，FAXマシンでは紙に書かれた白黒の情報をイメージセンサで読み取り，信号処理をして電話回線に送り出しています。イメージセンサで読み取る場合に画像をドット単位に変換するための変換誤差が発生し，特に横方向の線でギザギザになりやすい傾向があります。これは文字などでも同様に品位を落とす傾向にありますが，FAXモデムではこの現象は発生しません（図２参照）。

モデムとファックスの切り替え

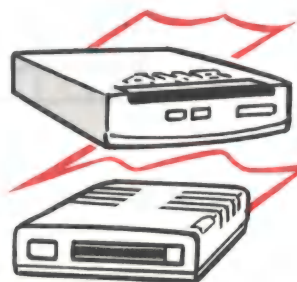
EIAの規格の項にあるように，ファックス部分の操作は，ヘイズ社のATコマンドにファックス機能設定用の拡張コマンドが追加され使用されています。ファックス用のコマンドは，大きく分類すると４つあります。

①DATAとFAXの切り替え
②FAXの送信データ速度設定
③FAXの受信データ速度設定
④フレーム間待機時間の設定

これらのコマンドは，ファックス通信ソフトウェアの中で既に組み込まれ構成されているので，直接ユーザーが扱う必要はありません。

特徴のあるファックスソフト

FAXモデムの機能を有効に利用するために，ファックスソフトウェアが用意されています。

アイワでは７月より高機能で使いやすいメガソフト社のSTAR FAX for AIWAの添付サービスを始めております。機能表は図３に示します。この中から代表的な機能を説明いたします。

●プリンタエミュレータ機能

一太郎，松などのワードプロセッサで作成したファイルをプリンタエミュレータを通すことによって，罫線や特殊な記号などがファックス用に変換され送信用ファイルが作成されます。プリンタで打ち出した場合と同等の画面でファックスに送り出すことができます。

●同報通信機能

会議の開催案内など同じ内容を複数の指定した宛先に自動送信ができます。

●イメージデータ送信機能

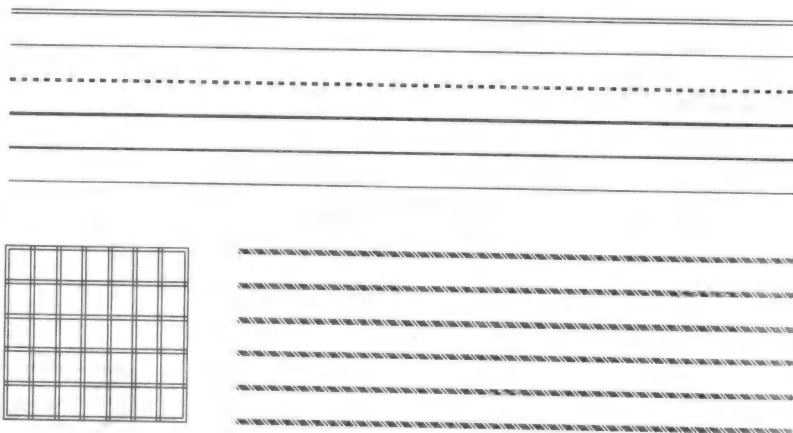
イメージスキャナから読み込んだデータをファックス信号に変換

ファックスモデム・送信テスト

AIWA FAX MODEM & STARFAX FOR AIWA

FAXMODEM: PV−AF144V5 −−− AIWA

ファックスモデムの送受信テストです。この原稿は㈱ジャストシステム社の一太郎のワープロにて作成したものをメガソフト社のスターファックスのプリンターエミレータにて処理して送信、NECのファックスマシンSPEAXにて受信したものです。

①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰⑱⑲⑳ Ⅰ Ⅱ Ⅲ Ⅳ Ⅴ Ⅵ Ⅶ Ⅷ Ⅸ Ⅹ ㍉㌔㌢㍍㌘㌧㌃㌶㍑㍗㌍㌦㌣㌫㍊㌻㎜㎝㎞㎎㎏㏄㎡

【図2(a)】FAXモデムの送信結果

して送り出すことができます。また，作画ソフトウェアなどで作成しイメージファイルも扱うことが可能です。

●**電話帳機能**

一項目の電話帳の内容はFAX番号，名称，氏名，などファックス電話番号＋5項目のメモで構成されています。トータル1万件の記録が可能です。この内容は，表紙に付ける送付案内や複数の送信相手先を指定する場合などに利用できます。

●**ファックス受信機能**

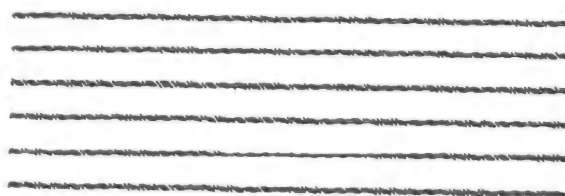
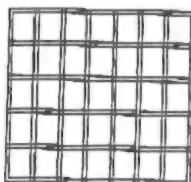
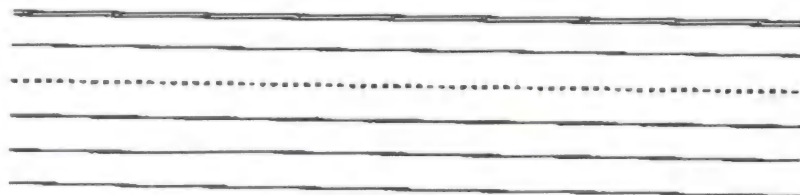
相手からのファックスを受信した場合，フロッピーやハードディ

ファックスモデム・送信テスト

AIWA FAX MODEM & STARFAX FOR AIWA

FAXMODEM: PV-AF144V5 --- AIWA

ファックスモデムの送受信テストです．この原稿は㈱ジャストシステム社の一太郎のワープロにて作成したものをメガソフト社のスターファックスのプリンターエミレータにて処理して送信、NECのファックスマシンSPEAXにて受信したものです。

①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰⑱⑲⑳ⅠⅡⅢⅣⅤⅥⅦⅧⅨⅩ ㍉
㌔㌢㍍㌘㌧㌃㌶㍑㍗㌍㌦㌣㌫㍊㌻㎜㎝㎞㎎㎏㏄㎡

【図2(b)】従来ファックスの送受結果(ノーマルモード)横方向のラインでギザギザが出る

スク上に受信データのファイルが作成されます。このファイルは画面に表示したり，プリンタに打ち出すことができます。１ページのデータ量は約50KBです。

アイワのFAXモデム4機種の概要

現在発売中の機種と，おのおのの特徴は下記のとおりです。

ソフト名：ＳＴＡＲＦＡＸ　for　ＡＩＷＡ　　(メガソフト㈱)
ＮＥＣ　ＰＣ９８パソコン用ＦＡＸソフトウェア

機能		メガソフト㈱ STARFAX V2.1	概要解説
環境	スキャナー対応	○	手書き原稿、図形をスキャナーで読み取り送信可能
操作	メニュー方式	○	メニュー画面から必要な操作が行える
	ﾊﾞｯｸｸﾞﾗｳﾝﾄﾞ処理	○	ワープロ等ｱﾌﾟﾘｹｰｼｮﾝｿﾌﾄから直接FAX送信可能
	ﾃｷｽﾄﾌｧｲﾙの表示	○	そのまま表示できる
送信可能データ	標準テキストファイル	○	ＤＯＳのテキストファイル
	ファックスファイル	○	受信したファックスのデータファイル
	ﾌﾟﾘﾝﾀｰｴﾐｭﾚｰﾄﾌｧｲﾙ	○	市販ソフトのプリントアウトイメージ
	イメージファイル	○	作画ソフトなどの画像データ
送信機能	同報通信	電話帳件数	同じ原稿を複数の宛先に送信
	時刻指定送信	○	指定した時刻に相手先に送信
	複数ファイルの送信	○	ファイル名の違う複数ファイルを同時送信
	送付案内	○	送信時に内容を編集作成可能
受信機能	自動受信	○	先方からのFAXを自動受信ファイル作成
	手動受信	○	電話機で確認後切り換えてFAX受信
	表示、印刷機能	○	受信原稿を表示、印刷が可能
電話帳	登録可能件数	10,000　件	ＦＡＸ番号、名前、メモ等の登録が出来る
	複数の電話帳管理	○	目的別に電話帳を作成できる
ＦＡＸ送信可能市販ソフト	一太郎 Ver3	○	プリンターエミュレーターの機能の動作をソフトメーカーが確認済みの市販ソフト
	一太郎 Ver4	○	
	松	○	
	P1.EXE／Note	○	
	オーロラエース	○	
	ＶＪＥ－ＰＥＮ	○	
	Z's STAFF Kid	○	
	Z's WORD JG	○	
	ＰＲＥＳＢＯＸ	○	
	アシストワード	○	
	花子	○	
	ＣＡＮＤＹ３	○	
	アートマスター400	○	
	LOTUS 1-2-3	○	
	桐　Ver3	○	
	マルチプラン	○	

【図３】
ファックス通信ソフト機能一覧

●PV-AF144V5

¥64,800

V.32bis 14,400bps超高速モデムとV.17 14,400bps高速ファックス機能搭載，AT Hayesコマンド＋V.25bis同期非同期モード対応，LCD大型表示器付きで，動作状態が確認できる小型高速モデム

●PV-AF24V5

¥29,800

V.22bis 2,400bpsモデム部＋V.29bis 9,600bpsファックス機能搭載，モデムの動作状態がひと目でわかるLCD大型表示器付き，スイッチング電源搭載小型ボックスモデム

●PV-AF24

¥26,800

電池動作可能な小型軽量ポケットサイズFAXモデム，V.22bisモデム部＋V.29ファックス機能搭載，MNP-10移動体通信用プロトコル採用，単3アルカリ電池にて8時間動作可能，ACアダプタ付き

●PV-CNF24

¥39,800

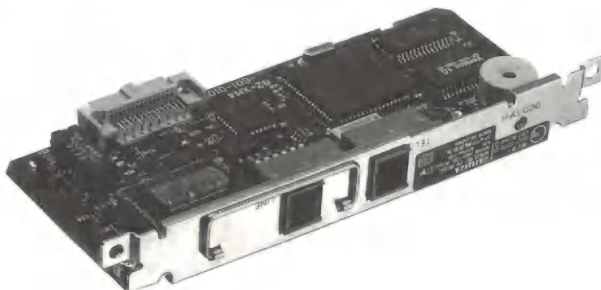
PC98ノート用カード型FAXモデム，V.22bisモデム部＋V.29bis 9,600bpsファックス部搭載，8MHz系クロックにもファックス動作対応

〈2,400bpsモデム
PV-AF24V5〉

〈2,400bpsポケットモデム
PV-PF24〉

〈2,400bpsモデム
PV-CNF24〉

オムロン

関 健彦

FAXモデムの概要

FAXモデムとは

FAXモデムとは，従来のモデムのデータ通信機能に加え，PC上で作成した文章・グラフィックスなどを電話回線を通じてFAXへ送信する機能を持ったモデムのことである。FAX自体が半二重モデムを搭載し，データ通信を行っているため，通常のモデムがFAXの変復調機能（V.29，V.27terなど）を搭載し，FAXモデムへと進化したのは当然の成りゆきといえる。

EIAでPCとFAXを接続するためのインタフェース標準が規格化されたあと，多数のFAXモデムの製品化が行われている。

サービスクラス

EIAによるPCとFAXモデム間の標準には，現在CLASS 1とCLASS 2があり，いずれもATコマンドを拡張したAT+F○○というコマンドを使用する。CLASS 3という標準も検討されているが，現在は規格化に至っていない。

各クラスの違いは，G3FAX機能（CCITT T.4，T.30）をPC（DTE），またはFAXモデム（DCE）のどちらで実現するかの違いであり，クラスが上がるほどFAXモデム側の負担が大きくなる。

CLASS 1では，PC側がFAX送受信に関してほとんどの手順をサポートする。そのため，CPU速度やOSの影響を受けやすいという問題がある。また，当然アプリケーション開発は，CLASS 2対応のほうが負荷が少ない。将来的にはCLASS 2，または3が一般的になると考えられるが，通常のFAX送受信に関しては，CLASS 1でも同じことが実現できる。

オムロンのFAXモデムに対する考え方

FAX通信アプリケーション

多くのFAXモデムが発売されはじめた今年の4月ごろまでは，市場に流通しているFAX通信アプリケーションがほとんどなかった。そのためか既存のFAXモデムでは，FAX通信アプリケーションをバンドルして発売している例が多い。

MD96XL10Vでも市場を喚起する必要からFAX通信アプリケーションをバンドルして発売しているが，基本的にはアプリケーションは，アプリケーションメーカーに任せハードウェアを市場に供給していきたいと考えている。

FAXモデムに関する考え方

当社では，300bpsからスタートしたモデムも現在では9,600，14,400bpsとなっている。形状も多様化し，エラー訂正やデータ圧縮も実現されパーソナルユースに関しては機能的にはほぼ十分なレベ

ルに達している。モデムのハードウェア単体での機能アップは限界に達しつつあり，ソフトウェアオリエンテッドな商品への転換も一方では必要であると認識している。

FAXモデムは，その転換を示唆する初めての商品であり，この商品を通じてアプリケーションメーカーと連携をとり，ソフト・ハードを合わせたトータルソリューションとしてユーザーに提供できる体制を整えることが必要である。

FAXモデム普及のための条件

現在の日本のFAXモデム市場は，まさに立ち上がり期にあり普及のためには，以下のような条件が考えられる。

①アプリケーション

- FAX機に勝る簡単なFAX送受信アプリケーションソフトウェアの創造
- バックグランドでFAX送受信処理を行うことができる環境の整備

②低価格化

- FAXモデムの低価格化
- パソコンの低価格化／高機能化

③標準化

- 標準化によるソフト／ハードウェアメーカーが完全に分業できる環境整備(標準化が一層進むことにより多くのアプリケーションメーカー，ハードメーカーが独立して参入でき市場が活性化する)

現在の状況は(マルチタスクを実現できる)Windowsの普及，CLASS 2の規格化などがあり，FAXモデムの普及に関しては追い風がふいているということができる。

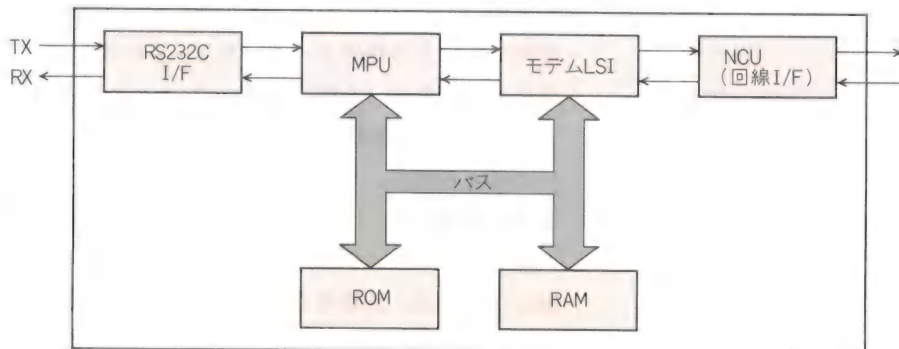
FAXモデムの今後

FAXモデム自体はV.17，V.FASTへの高速化，ECMへの対応，FAX通信アプリケーションへのOCR機能搭載などによる高機能化が考えられる。

大きなトレンドとしては，DATAモデム→FAXモデムに続き音声(VOICE)という方向が考えられる。音声を扱うようになると，一層ソフトウェアオリエンティドな対応が必要となる。すでにモデムLSIには音声をハンドリングする機能が盛り込まれているものもあるが，今まで音声を扱ってきた家電製品に対し，音声の加工などの点で十分な魅力を持ったアプリケーションが出現する必要がある。

MD96XL10Vの技術的ポイント

MD96XL10Vは，当社が6月25日より発売しているデータ通信

【図1】
MD96XL10V
ブロック図

9,600bps, FAX通信9,600bpsの小型FAXモデムである。

以下のような点を考慮し，設計，商品化を行った。

ブロック図を**図1**に示しておく。

①携帯可能モデム

MD96XL10Vは，当社では初めての携帯可能な9,600bpsモデムである。携帯可能モデムは1,200pbsからスタート，現在では2,400bpsまで商品化しており，2,400bpsパーソナルモデム市場の半分近くを占めていると見られる。

携帯可能モデムは，実際に多数のユーザーが携帯して使用しているとは考えていない。一般的にはノート型パソコン同様，机上スペースの有効活用の意味を持っていると思われる。その意味で，MD96XL10Vの形状は小型でありながら据置も考慮し，さらに縦置き，横置きいずれでもLEDの視認性が良い形状とした。

高速化と合わせ小型化(低消費電力化)もモデムの向かう1つの方向である。小型化のために以下のような実施策を取った。

- 9ピンDTEコネクタの採用(ただし，変換コネクタを同梱しMACやDOS/V機も対応)
- 小型ライントランスの採用
- LEDも含めたチップ部品の多用

9,600bpsの携帯可能なデータモデム，FAX／データモデムは現在国内でもいくつかのメーカーから発売されており，今後2,400bpsのモデムと同様に市場を得ていくものと考えられる。

②低消費電力化

携帯型のモデムを考えるとき問題となるのは，消費電力である。当社の2,400bpsの携帯可能データモデムが発売された当初，006P型マンガン乾電池で公称1時間の連続使用が可能であった。

今回9,600bpsモデムを開発するにあたり，マンガン電池の連続使用時間のMIN値として同様1時間の連続使用を設定した。

モデムの消費電力のうち大半を占めるのがモデムLSIである。

MD96XL10Vでは，新しい低消費電力タイプのモデムLSIを使用することによって消費電力を下げることが可能になった。ほかにもフォトモスリレーの使用，LEDの電流値低減(ケース形状を含めた輝度確保)，ICすべてのCMOS化などで低消費電力化を行った。近い将来，3.3V駆動のモデムLSIの出現により，さらに低消費電力化が実現される。

③MNP10

エラー訂正の手順であるMNPクラス10を搭載した。MNPクラス10は，移動体通信のような品質の悪い回線での通信を目的としたエラー訂正手順であり，回線の切断，パフォーマンスの低下などを防ぐものである。基本的な動作は，MNP通信中のパケットサイズの変更と通信速度の変更である。そのため多くの通信速度に対応した高速モデムほど有効な機能ということができる。

実際にNTT7リンク相当の非常に品質の悪い状況での回線保持について有効であり，携帯して移動体通信に使用する，しないにかかわらずメリットがあるものと判断している。

④FAXクラス1

1項目のFAXモデムの概要のところで述べたとおり，今後CLASS

2が主流になると思われるが，MD96XL10VはCLASS 1対応である。開発時点でCLASS 2は規格化されておらず，また暫定的なCLASS 2を採用することもできたが，CLASS 2規格の行方が明確でなかったこと，暫定の規格を採用することにより，最終的に市場の支持を得られない可能性があると考えたからである。

結局，現時点では市場に多くの暫定CLASS 2 FAXモデムが存在しているが，対応するFAX通信ソフトも多くの暫定，または独自規格が存在する米国のものを日本語化したものが多数であり，大きな問題とはなっていない。

⑤PC9800シリーズ対応

国内でFAXモデムを商品化するときに最大の問題点となったのがこの点である。EIAの標準では，FAX通信を行う場合のPC-FAXモデム間のスピードを19,200bpsとしているが，PC9800シリーズのCPUクロック8MHz系ではその速度を実現できない。この件に関してはRockwell社（モデムLSIメーカー）から示された，RS-232Cとしては標準的でない20,800bpsという速度に対応することで解決が図られている。現在の国内FAXモデムの多くが，その方式を使用していると思われる。

MD96FL10Vでは使用しているMPUに，クロックジェネレータがない。そのため，VCCI対策はきつくなるが，G/Aの空きを使用した分周で20,800bpsに対応している。

なお，FAX通信ソフトも98シリーズに対応できるものを標準添付しており，一太郎のような市販のアプリケーションソフトで作成した文書を相手のFAXへ送信できる機能もある。

最後に，参考に今まで述べた内容を表と図に示しておく。

以上，簡単であるが当社のFAXモデムの考え方と技術ポイントについて述べた。

CLASS	PC（DTE）	FAXモデム（DCE）	規格番号
CLASS1	T．4 T．30		578
CLASS2	T．4	T．30	592
CLASS3		T．4 T．30	

CCITT　T．4　：G3FAXの端末特性，符号化方式
　　　　T．30：FAXの伝送制御手順

〈各クラスの概要〉

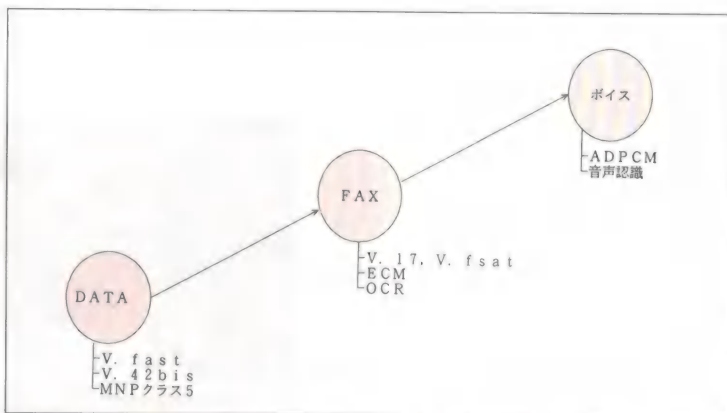
データ通信	FAX通信
MNPクラス5 V．42bis	クラス2FAX
MNPクラス4 V．42	クラス1FAX
AT	AT+Fコマンド
コマンド	
モデムLSI制御ソフト	

↕

モデムLSI

〈FAXモデム ファームウェア概念〉

〈モデムから見た機能トレンド〉

③

メガソフト

内海　徹

日本のパソコンFAX の代名詞 STARFAX

STARFAXは，日本のFAXモデムの代名詞的存在であり，1987年に初めてリリースされている。

初期のSTARFAXは，「STARFAX，MODEL20」，「STARFAX MODEL7」というPC-98シリーズの拡張スロットに挿入するボードタイプのパソコンFAX(全二重モデムは搭載していない)であり，ハードウェアにディペンドしたSTARFAXソフトが標準添付されている。

現在もSTARFAXシリーズとしてハードウェア／ソフトウェアとも，多くの製品が発売されており，STARFAXファミリーとも言える製品群となっている。

STARFAXシリーズの特徴

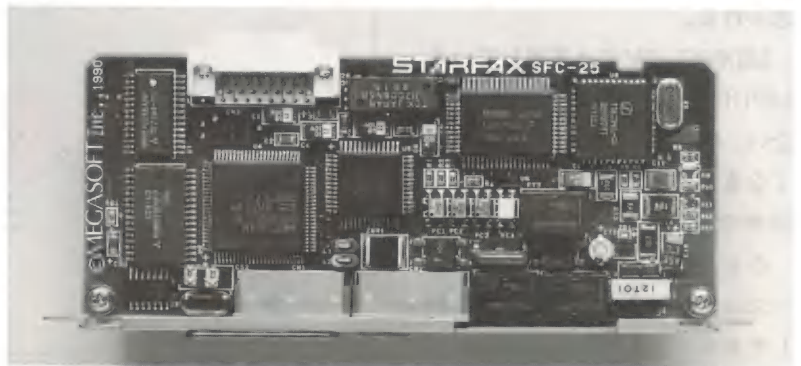
(1)STARFAX MODEL20

PC-98シリーズのデスクトップタイプの拡張スロットを利用する内蔵ボードタイプであり，現在のSTARFAXの基礎を築いたモデルである。

開発時期が古いため全二重モデムは搭載していないが，FAXの送受信という点で考えると安定性バツグンのモデルである。

現在のSTARFAX MODEL20に付属しているソフトは，STARFAX MODEL25やSTARFAX9624マルチパックに添付されているソフトとは異なる古いタイプのものだが，新しいタイプのソフトを利用する場合は，STARFAX Ver2.1を別途購入すれば新しいSTARFAXとして使用できる。

利用できるフォントは，16ドットおよび24ドットフォントだが，一太郎などのソフトから直接FAXを送る場合は，一太郎で利用できるフォントが使用可能である。

1987年当時，STARFAX MODEL20が画期的だったのは，G3 FAXに直結してプリンタ／スキャナとして使える機能，および，STARFAX間での高速ファイル転送機能であるが，これらは以降の

〈写真1〉 STARFAX NOTE

全STARFAXシリーズでも継承されている(ただし,メガソフト製のハードウェアのみ)。

(2)STARFAX NOTE

STARFAX NOTEは,98NOTEの拡張スロットに内蔵できる非常にコンパクトなFAXモデム(**写真1**)であり,全二重2,400bpsモデムも内蔵している。このため,98NOTE 1台でパソコンFAXもパソコン通信も可能としている。

NOTEタイプの場合,電源やRS-232Cが不要な内蔵型FAXモデムは,本体の中に隠れてしまうので携帯性の面で非常に便利である。

STARFAXソフトが標準添付されており,MODEL20同様,16ドット,24ドットのフォントが使用できる。

一太郎などのプリンタエミュレータ機能も同様に使え,今後このタイプはPCMCIA2.0/JEIDA4.1のカード型FAXモデムに形を変え,いままで以上に普及することが予想されている。

(3)STARFAX 9624マルチパック

STARFAX 9624は,携帯型のコンパクトな外観を持つ外付けタイプのFAXモデム(**写真2**)である。内部的には国際規格であるCLASS 1 (EIA/TIA578)仕様であり,標準添付ソフトもCLASS 1対応ドライバを使用している。国際規格であるので,STARFAX 9624をほかのWindows,もしくはMacintoshなどのFAXソフトで使うことも可能である(ただし,メーカーの保証はない)。

STARFAX 9624マルチパックには,DOSでPC98シリーズ用とDOS/V用のSTARFAXソフトが標準添付されており,さらに,話題のWindows対応のドライバも標準添付されている。

なお,今後の予定として全二重通信14,400bps/FAX14,400bpsに対応した高速バージョンの発売も予定されている。

(4)STARFAX Ver2.1

STARFAX Ver2.1は,CLASS 1対応のSTARFAXソフトウェアである。PC98シリーズおよびEPSONシリーズ対応であり,サポート対象ハードウェアはEPSON MX-240 FAXモデムである。

機能的にはほぼSTARFAXシリーズと同等であるが,高速ファイル転送やリモートFAX機能など,STARFAX独自の機能は使えな

〈写真2〉
STARFAX 9624

い。

今後はMX-240だけでなく,続々と増える他社製FAXモデムのサポートも拡張していく予定である。

(5)STARFAX SFC-30A

SFC-30A(**写真 3**)は，STAR FAXシリーズの豊富な経験をもとに,業務用FAXボードとして開発された製品であり,SHOP売りなどは行っていない。

音声機能とFAX機能を標準で搭載しており，このため，今後大きな市場になると予想されている音声FAXサービス市場をターゲットとしている。

SFC-30Aの最大の特徴は，マルチ動作が可能という点であり，このため，複数回線対応の音声FAXサービス用途をはじめ，今後，オフィスで不可欠になると予想されているFAXサーバー用のFAXボードとしての需要が見込まれている。

(6)STARFAX EXPRESS

STARFAX EXPRESSは，SFC-30Aをベースに24時間自動音声FAXサービスシステムとしてメガソフト自身が開発したシステムであり，購入したその日から24時間対応の音声FAXサービスが運営できるようになっている。複数回線対応となっており，必要に応じて1回線ずつの増設が可能である。

なお，STARFAX EXPRESSは，24時間デモサービスが行われているので,興味ある方は03-5213-6736(東京)，06-337-1767にFAX機から電話してみるとよいだろう。

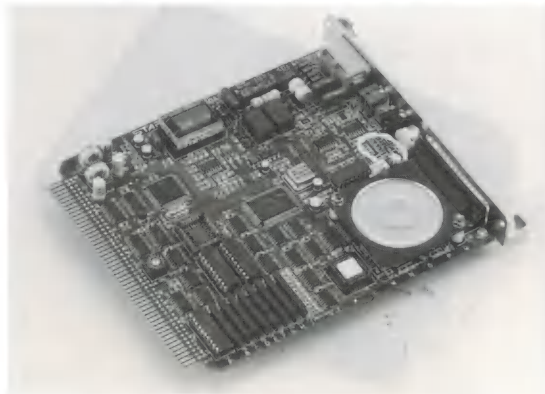
すぐに超鮮明なFAXが送られてくる。

STARFAX独自のリモートFAX出力機能

他社にない,STARFAXの特徴の一つが，このリモートFAX機能である。

この機能はSTARFAXをG3FAXに直結し,G3FAXそのものをプリンタとスキャナとして使える機能である。どこにでもあるG3FAXをプリンタの代わりとして使えるため，海外など日本語プリンタのない環境でも簡単に資料をプリントアウトすることができ，便利である。

また,G3FAXをスキャナとしても利用できるので，貴重な資料をG3FAXから画像ファイルとしてノートパソコンに入れて持ち帰ることもできる。

STARFAXどうしで可能な高速ファイル転送機能

〈写真 3〉
STARFAX SFC-30A

これもまた他社にはない(INTEL SatisFAXtionを除く)特徴であるが，メガソフトがハードウェアとソフトウェアとペアで発売しているSTARFAXシリーズには，すべて9,600bpsでファイル転送ができる機能が装備されている。

通常のパソコン通信用モデムを使ったファイル転送はかなり面倒であるが，STARFAXのファイル転送では，受信側はソフトを起動しているだけで自動で受信できるので非常に手軽にファイル転送が可能である。一度の送信で最大50個までのファイルを転送できる。

STARFAXのプリンタエミュレータ機能

STARFAXではSFEM，EXEという小さな常駐ソフトをアプリケーションソフトを起動前に常駐させることにより，DOS上のほとんどのアプリケーションソフトの印刷イメージそのままをFAX送信できるようになっている。

例えば一太郎であれば倍角や袋文字はもちろん，花子から取り込んだ図形なども入れて，FAXファイルを作ることができるし，メモリーさえ許せば一太郎上からでもFAX送信が可能である。

STARFAX for Windows

STARFAX9624マルチパックには，STARFAXのWindows用FAXドライバが入っている。

このドライバを使用することでWindowsのアプリケーションソフトからダイレクトにFAX送信が可能である。

当然，漢字TrueTypeのスケーラブルなアウトラインフォントが使えるので，DOSに比べて非常に表現力あるFAX送信が可能となっている。

なお，現在のSTARFAX for Windowsは，まだ単体のFAXソフトではなく，STARFAX9624マルチパック，そして今後登場する予定のSTARFAX1414マルチパックに同梱されて出荷される見込みである。

安くなるSTARFAXシリーズ

先日のNECパソコンフェアでも紹介されていたが，STARFAXシリーズは，8月末あたりにSTARFAX1414が発売され，既存のSTARFAX9624なども値下げをするようである。

FAXモデムの価格は，この1年間に急激に降下しており，今後も下がることが予想されている。

8月末から値下げする製品と価格は下記のとおりである。

- STARFAX1414
  定価79,800円
- STARFAX9624
  定価59,800円
- STARFAX NOTE
  定価49,800円

④

ソニー

岩瀬 純夫

ソニーFAXモデムのコンセプト

当社では，電子メールダイレクト送受信や掲示板サービスができる統合型ネットウェアにFAX送信機能を盛り込んだメモリーFAXモデム(SMD-50/70)を販売しています。

FAXの受信は実際に使われることは少なく，使われない機能にコストをかけるのは不合理的であるとの考えで，むしろ送信をより便利(快適)に活用できるように「パソコン不要のFAX送信」をコンセプトに商品化しました。

FAXモデムで問題と思うこと

FAXモデムはパソコンで操作するので「大きな画面で，しかもマウス操作も可能」です。そのためか一般のFAXよりも同報送信や時間指定送信が楽に設定できこの機能を利用される方も多く，そのために逆にクローズアップされる大きな問題があります。

それは，次のようなものです。

- FAX送信中は，パソコンが使えない(特に同報送信時は長時間使えない)
- FAXをバックグラウンドで送れるが，その間メインのソフトが遅くなる
- 時間指定送信をする場合それまでの間，パソコンの電源が落とせない
- 現在送信中の場合は，別の場所に別のFAXを送る登録ができない

メモリーをFAXモデムに搭載すれば

これらの問題解決が特に重要との認識のもとに，メモリーFAXモデムを開発しました。送信先が遠距離の場合は，時間指定しないと通信費が大きな負担になります。次に時間指定した場合と，そうでない場合の通信費を比較してみました(**表 1**(A4原稿12ページの例))。

また，一か所に直接FAX送信する場合でもメモリー機能を利用すればパソコンの拘束時間は短くて済みます。A4で12ページの原稿を送った場合を実測してみました(**表 2**)。

このようにパソコンを解放し，SMD単独でFAX送信ができます。送信登録できるFAXの原稿は，いわゆる標準原稿でSMD-50(1MB実装)の場合はA4で50ページ，SMD-70(4MB実装)の場合は200ページが目安になります。

資料 送信ﾓｰﾄﾞ	昼間ﾀﾞｲﾚｸﾄ 送信(8-19)		時間指定送信 深夜(23-7)		時間指定で 節約できる金額		年間(100通)で 節約できる金額	
通信ｴﾘｱ	国内	米国	国内	米国	国内	米国	国内	米国
通信費	650円	1828円	350	1,163	300	665	30,000	66,500

【表1】
通信料金比較
(93年5月10日
現在：国内は最遠
距離320km以遠)

ソニーFAXモデムの その他(FAXを除く部分)の機能

ソニーのFAXモデムSMD-50(39,800円), SMD-70(59,800円)には,FAX以外にも魅力的な機能を満載しています。次に,それらをご紹介しましょう。

(1)メール着信が即座にわかる"留守番モデム"機能

留守中にメールを受信すると,本機のメモリーに格納されると同時にインジケータが点灯しメールの着信を知らせます。今までのようにパソコンを立ち上げてネットにアクセスするまで,メールの着信がわからないといった不便さが解消され,緊急の通信を受け損なう恐れがなくなりました(FAXからの直接受信はできません)。

(2)1本の電話回線で「電話/FAX/データ通信」を自動切り替え

1本の電話回線で電話(留守番電話を含む)/FAX/データ通信を自動的に切り替えます。本機のために専用回線を引く必要はありません。

★接続できるFAXは無鳴動着信ができる機種に限られます。(G3機/1,300Hz対応)

★相手FAXが手動発信時,CNG信号を出さない機種は,この機能を利用することはできません。

★留守番電話機で応答メッセージが(音声合成など)録音できない機種は,この機能を利用することはできません。

(3)通信コストを大幅にセーブする「自動対話サービス機能」

①商業ネットを介することなく,ダイレクトに個人間でメールのやり取りが可能です。商業ネット加入の場合と比べてネット加入料,アクセスチャージなどが不要です。

②世界最小のパソコン通信ホスト局が開局できます。低消費電力(約4W)で,しかも,パソコンやワープロを動作させておく必要がないので,電気代の節約になります。

③自前のネットなのでメールのセキュリティーは万全。安心して情報のやり取りができます。

④BBS(掲示板)機能を持っているので,アクセスしてくる人が自由にメッセージを書き込んだり読み出したりすることができます。

ソニーモデム用の FAXソフト最新動向

①PC-98シリーズ

SMD-Talk98というSMD専用の統合型のソフトをソニーから8月初旬に発売する予定です。価格は1万円強になる見込みです。このソフトを使えば,SMDのメモリー機能をフルサポートしたFAX送信が可能です。

送れるデータは,DOS上のアプリケーションおよびWindowsアプリケーションで印刷できるものの両方をサポートしています。また,メモリー送信時も宛先別のFAXカバー(FAX送付表)を付けることが

【表2】パソコン拘束時間比較

資料の送信時間	直接送信	メモリー送信（登録時間）
A4　12ページ	9分45秒	6分30秒

できます。

ほかのFAXソフトもありますが，メモリー機能をサポートしていないので，パソコンから完全に切り放したFAX送信はできません。操作性などSMDを前提に作られているソフトですので，PC-98および互換機ユーザーにはSMD-Talk98をお勧めします。

また，12月にはSMD用プリンタドライバ形式のWindows上で動作するFAX専用ソフトを発売する予定です。

②IBM-PC

WindowsV3.1環境で動作するWinFax PRO V3.0やWinFaxライトを利用することができます。モデムの設定は，米国メーカーのいくつかのSEND FAX機種が登録されていますので，その中から適当な機種を選べば動作します。

SMDにはシェラのFAXチップが搭載されています。米国製のFAXソフトにはシェラ搭載機種が多く登録されていますので，SUPER-FAXやBitFaxなども未確認ですが，FAX速度9,600bpsのSEND FAX機種を選べば同じように使えるものと思います。

なお，ソニー製のSMD専用のWindowsFAXソフトは12月に発売する予定です。

③MAC

10月にメモリー機能をサポートしたFAXソフトを発売する予定です。プリンタドライバ形式で，簡単な操作でMACのグラフィックスアプリケーションも含め，FAX送信が可能です。

送信先が複数指定されると，「速達」，「コストセーブ(自動タイムシフト)」，「マニュアル(時間設定)」のスイッチがあらわれ，宛先に遠方個所も含まれる場合は，「コストセーブ」を選べば自動的に深夜料金で相手に送れます。メモリー送信ですので，その間MACに負担を与えることは皆無で，パソコンの電源を完全に落とすこともできます。

また，複数台のMACでSMDを共有できるネットウェアの開発も進めています。

ソニーFAXモデムの技術的ポイント

ソニーのFAXモデムは，メモリー機能を評価いただき選ばれると思いますので，メモリー機能を利用したFAX送信の具体的なテクニカルインフォメーションを提供させていただきます。なお，一般に利用いただく方は，それぞれの機種(パソコン)に対応した便利なソフトを準備していますので，それを利用いただけます。

メモリー送信を実行するには

(1)SMD用FAXファイルを用意する

パソコンで画データを作り，それをFAX特有のMH圧縮加工したファイルを用意します。そのファイルの先頭にアスキーでSMDFAX10のヘッダーを付け加えます。

(2)SMDのメモリーにFAXファイルを書き込みます

SMDには3つのドライブに相当するRAMディスクを装備しています。それらのドライブには，次の3つの名称が付与されています。

その中でFAX送信用のファイルを格納するドライブとして，③のATSMALを推奨します。

①ATBBS
②ATRMAL
③ATSMAL

これらのドライブはファイルハンドラで管理され，書き込みや削除を繰り返してもガベージコレクションのような問題は発生しません。また，作成されたファイルは，回線側からも232C側からも同時にアクセス可能なようにSMDはマルチタスクOSを持っています。

DTEとのインタフェースは最大19,200bpsをサポートします。用意したFAXファイルは，次の方法(コマンド操作)でSMDに書き込みます。

ATSMAL/W filename(max 8 bytes):BX1(OR BX2):::

DTE側からSMDに上記コマンド文字列を与えると，ファイルのバイナリアップロードが可能な状態になるので，XMODEMまたは無手順でFAXファイルをアップロードします。

ファイル名の次のパラメータがアップロードプロトコルオプションで，BX 1を指定すると128SUMのモードで，BX2を指定すると1024 CRCのモードで直ちにアップロードすることができます。ここで，無指定(:BX2:::→::::)は，バイナリ無手順のモードでのアップロードになります。

バイナリ無手順の場合は，アップロード終了をタイムアウトで見ています。そのタイムアウト時間を制御するのは，次のコマンド(ATSYSの第4レジスタを操作)によります。制御可能時間範囲は3－15秒であり，初期値は5(秒)です。

例1：バイナリ無手順アップロードタイムアウト時間を3秒にする場合

ATSYS/S 4:3

FAXファイルをATSMALドライブに書き込んだら，ATSMAL/？でドライブ内のファイル一覧を見て正しく書き込まれたかどうかを確認します。ATSMAL/？は，MS-DOSのDIRと同じ機能であり，1行の左8バイトがファイル名です。

(3)送信実行登録をする

SMDではヘイズのダイヤルコマンドATD(P/T)を大幅に拡張したATSENDコマンドで，次の機能をサポートしています。

①送信先FAX番号登録(1コマンド操作当たり最大255か所)
②各宛先への送信時間指定
③送信ファイルの指定(最大10ファイル以内)

また，SMDのシステム管理としてATSENDコマンドとは別に，ATSCON コマンドで「話中(BUSY)」時の再送回数，再送間隔を設定することができます。

図1に，具体例で説明します。

(4)再送間隔，再送回数を設定する

SMDはシステム管理コマンドATSCONで，再送信間隔と再送回数を設定することができる。

例1：BUSY時，再送信間隔を10分に，再送信回数を5回に設定す

る。

ATSCON/S :5:10::::::::::::::

(5)送信結果を参照する

送信結果をSMDから読み出すには，次のコマンドを操作します。**図2**に，コマンド操作したときの画面イメージ例を示します。

ヘッダー以降の各行がFAX送信宛先情報と状況を現します。正常終了の場合はOKを表示し，指定された再送信回数実行した結果もエラーの場合は，そのエラーとなった原因をエラーメッセージとして帰します。

主な仕様

- 使用回線
  一般電話回線，PBX内線(使用できない場合があります)
- 制御コマンド
  ヘイズ／マイクロコムATコマンド
  ソニー拡張コマンド
- 通信方式
  全二重CCITT V21/V22/V22bis準拠
- 通信速度
  300/1,200/2,400bps
  (SMD間最大スループット

```
例１：ＦＡＸ番号03-1234-5678の山田さんへFAXFILEという原稿のＦＡＸを送る

ATSEND/W SEND0001::::FAXFILE:F0312345678


例２：ＦＡＸ番号03-1234-5678の山田さんと06-555-5555の鈴木さんへFAXFILEという原稿
のＦＡＸを送る

ATSEND/W SEND0002::::FAXFILE:F0312345678/F065555555


例３：ＦＡＸ番号03-1234-5678の山田さんと06-555-5555の鈴木さんへFAXFILE1とFAXFILE
2という原稿のＦＡＸを送る

ATSEND/W SEND0002::::FAXFILE1/FAXFILE2:F0312345678/F065555555


例４：ＦＡＸ番号03-1234-5678の山田さんと06-555-5555の鈴木さんへFAXFILE1とFAXFILE
2という原稿のＦＡＸを送る。なお、鈴木さんへは深夜(23:30)に送る

ATSEND/W SEND0002::::FAXFILE1/FAXFILE2:F0312345678/F065555555<T2330>


例５：１０００ヵ所へＦＡＸ同報送信をする

ATSEND/W SEND0001::::FAXFILE:F[tel number0001]/F[tel number n]/････F[tel number0
255

ATSEND/W SEND0002::::FAXFILE:F[tel number0256]/F[tel number n]/････F[tel number0
512

ATSEND/W SEND0003::::FAXFILE:F[tel number0513]/F[tel number n]/････F[tel number0
768

ATSEND/W SEND0004::::FAXFILE:F[tel number0769]/F[tel number n]/････F[tel number1
000


例６：東京から北海道のＳＭＤを中継し、北海道札幌市内３０ヵ所にＦＡＸを送る。なお、
東京から北海道へは深夜２３：３０に送信をし、札幌市内には東京からの送信完了後ただ
ちに配信を開始する。

ATSEND/W SEND0001::::FAXFILE:011-XXX-XXXX<Uxxxxxxxx><Lxxxxxxx><Pxxxxxxxx><T2330>
+F[tel number 1]+F[tel number n]+ ･･････ +F[tel number 30]
```

【図１】
実例

9,600bps)

- 端末速度(PC←→SMD間)
  最大19,200bps
- 通信プロトコル
  無手順/XMODEM
  128SUM/1024CRC
  MNP4/5/V42/V42bis/専用モード
- 変調方式
  FSK(300bps)
  PSK(1,200bps)
  QAM(2,400bps)
- キャリア周波数　300bps
  1,200／2,400Hz
  ORG側
  スペース：1180+/−6Hz
  　　　　　1200+/−.5Hz
  マーク：　980+/−6Hz
  ANS側
  スペース：1850+/−6Hz
  　　　　　2400+/−.5Hz
  マーク：　1650+/−6Hz
- ダイヤル信号
  パルス(10/20PPS)，トーン
- NCU　MM型，MA型，AA型
- CPU　10MHz 64180
- 内部メモリー
  SMD-50　約1MB
  SMD-70　約4MB
- FAX機能
  G3送信機能
  伝送速度：
  9,600/7,200/4,800/2,400bps
  伝送時間：
  約20秒(標準モード,標準原稿)
  伝送手順：バイナリー手順
  符号化方式：1次元MH符号化
  主走査線密度：8ドット/mm
  副走査線密度：
  ノーマル：3.85本/mm
  ファイン：7.7本/mm
  (コマンドにて選択可)
- 電源
  AC100V 50/60Hz
  (電源アダプタ使用)
  メモリーバックアップ用DC
  6V(1.5V×4)アルカリ乾電池
- 消費電力
  SMD-50
  通信中　4W，待機時　3W
  SMD-70
  通信中　6W，待機時　4W
- 主な付属品　電源アダプタ
  RS-232Cケーブル
  PC-98シリーズ用FAX送信ソフト
  PC-98シリーズ用フリーソフト集
- 寸法：約120×152×40mm
  (幅×奥行き×高さ)
- 重量：約0.4kg

【図2】
画面イメージ例

```
ATSSTS/?

送信 ﾒｰﾙ                                                   発信手続
   ﾌｧｲﾙ  C  発信完了時刻  発信先電話番号      ﾕｰｻﾞｰID    容量    ﾌｧｲﾙ   状況
19180259 M 93/05/19 18:06 03-3375-7518       START000  34816 19180259  OK
19180732 M 93/05/19 19:06 03-3375-7518       START000 562560 19180732  OK
TEST1B   M 93/05/25 14:09 F3458-7160         START000  14336 SEND      OK
TEST1B   M 93/05/25 14:15 F3458-7160         START000  14336 SEND      OK
TEST1B   M 93/05/25 14:18 F3458-7160         START000  14336 TEST      OK
TEST1B   M 93/05/25 14:58 F03-3458-7160      START000  14336 SEND      OK
TEST3A   M 93/05/25 16:20 F03-3458-7160      START000  44032 SEND     N-BUSY
TEST     M 93/05/25 17:32 F3458-7160         START000 130048 TEST      OK
TEST     M 93/05/25 17:49 F03-3458-7160      START000 130048 SEND      OK

OK
```

FAX通信ミニセミナー

谷沢 俊昌

FAXモデムとFAX機、およびソフトとの互換性

よく、「家のFAX機ではA君の所にはFAXが送れるが、B君の所には送れなくて困る」といった話を聞きますが、同じG3規格のFAXでもFAX機の製造時期の違いや製造メーカーの違いでFAXを送れない、受け取れないといった不具合があるようです。

FAXモデムで送信した場合も同じことが言え、AメーカーのFAXモデムは、α君の所のFAX機にはFAX送信できるが、β君の所にはFAXが送れない。

ところが、BメーカーのFAXモデムは、α君の所にもβ君の所にもFAXが送れるといった例もあります。FAX機もFAXモデムも、同じG3規格に準拠して作られているのに、どうして……？

G3規格に準拠しているわけですから、手順やコマンドなどは同じと言えますが、タイミングの設定などには製造メーカー独自の考え方が反映されている場合がありますから、完全に合致するとは考えられません。したがってBメーカーのFAXモデムが、α君の所にもβ君の所にもFAXが送れたのは、BメーカーのFAXモデムのタイミングの設定のラチュード(寛容度)が広かったから両方のFAX機とタイミングが合ったものと考えたほうがよいようです。このことはFAX通信ソフトにも言えることで、FAX通信ソフトは、パソコンとFAXモデムを結んで、相手のFAX機やFAXモデムにFAX機で送るのと同じ感覚でデータを送る役目をしておりますから、パソコンとFAXモデムを相性良く結び付けるものでなくてはなりません。

それと、通信エラーも起こさず正常に終了した場合でも、相手側ではノイズが混入して文字化けや文字落ちが生じる現象もありますので、送信後に連絡をとり正常に送られているかを確認する必要があります。相性が良いからといっても、回線状態の変化でいつこのような現象が起きるかわからない不安定さも考慮しておかないと、とんだ失敗を招く恐れもあります。

汎用FAX通信ソフトとして開発されている場合でも、開発側は特定の機器を対象に開発していますから、開発側の環境と合致していない機種で使用した場合には、CPUのパワーや速度の違いから互換性を保証できない場合も当然起こります。

よく言われることですが、今まで使っていたハードディスクでは正常にFAXが送れたのに、ハードディスクを高速なものに換えたらFAXが送れなくなってしまったとか、FEPを変更したら駄目になったとか、ドライバを変更したらエラーが出たなどのトラブルに遭遇することもあります。

特に98の通信環境でFAX通信を行う場合には、問題点が多くあって「98ではFAX通信は無理だね」などと言われる方も多いようです。

Windows用のDOS/V版FAX通信ソフトの日本語化は完成したものの、そのままでは98Windows上では機能を発揮できないとか、Windows上ならDOS/V版でも98で使えるものもあるとか……98ユーザーからさまざまな声が聞かれます。

しかし、駄目だからといって放っておいては、FAXモデムの日本での普及は望めないと思いますし、98本体の通信環境の悪さをカバーして、共存共栄をといった周辺機器メーカーの動きも盛んになってきました。

古い言葉で言えば「必要は発明の母」、モデムチップメーカーROCKWELLが、8MHz系のCPU搭載機でもFAX通信ができるように、20,800bpsに対応したチップを開発して日本のモデムメーカーに提供したことにより、日本のFAXモデムは今後

このチップを搭載して発売されるものが多くなり、日本のFAXモデム界にも活気が出て来るものと思われます。アメリカより2〜3年は遅れていると言われる日本の通信事情ですが、今年後半になってFAXモデムやFAX通信ソフトの発売の動きも活発になり、FAX情報時代、FAX通信時代の本格的な幕明けが感じられるようになりました。

国産のFAXモデムやFAX通信ソフトの普及により、それら相互の対応合戦により、日本のFAX通信事情も好転するものと思われますし、日本の環境に合致したニューメディアとして、FAXモデムやFAX通信ソフトの互換性も向上して行くものと期待しています。

FAXモデムの選び方使い方

従来、モデムといえばデータ通信に使用する周辺機器でしたが、FAX機能を搭載するに及んで、FAXモデムと呼ばれるようになったため、データ通信専用のモデムはデータモデムと呼ばれるようになってきました。

とはいっても、FAX機能はあくまでもデータモデムの付加機能であり、「FAX付き」とか「FAX機能搭載」といった形で扱われており、従来型のデータモデムと並行して発売されています。

2,400bps、9,600bpsのデータモデムを持っている人が、さらに高速な通信環境を望んで14,400bps対応のモデムに買い替える際に、「どうせ買うのなら先行きのことも考えてFAXモデムを買っておこう」とFAXモデムを買う場合もあります。

また、「FAX通信をしたいだけなら、9,600bpsや14,400bpsのFAXモデムで十分」といった見方をする人もいます。

高速データ通信を行わないのなら、2,400bps FAX機能付きでも14,400bpsFAX機能付きでもFAX機能に変わりはないので、そういった選択も考えられます。また、G3規格のFAXの通信速度にも9,600bpsと14,400bpsがあって、これも選択の条件に入りますが、会社とかオフィスなどに設置されている14,400bps対応のFAX機を対称にFAX送信するのでなければ、現在使用されているFAX機のほとんどが9,600bps対応となっているので、9,600

bpsでも十分役に立つものと思われます。この通信速度も将来のことを考慮して、14,400bps対応のFAXモデムにといった人もいますが、その時代が到来するのはいつなのかを予測することは困難なように思われます。

また、国内で発売されているFAXモデムには、CLASS1対応とCLASS2対応のものがほとんどですが、これにもCLASS1だけ、CLASS2だけ、それにCLASS1とCLASS2両方に対応しているものがあります。

FAX通信ソフト面でも対応しているCLASSが分かれていますが、FAXモデムにバンドルされている通信ソフトはほとんどがCLASS1対応のDOS用となっていますし、国産のFAX通信ソフトもDOS用Windows用いずれも現時点ではCLASS1対応となっています。外国産のFAX通信ソフトには両クラスに対応しているWindows用のものもありますが、日本語版は数も少なく手に入ったとしてもDOS/V版ですので、98Windowsでは動作しないものもありますので、国産のFAX通信ソフトがCLASS2に対応するのを待ったほうが良さそうです。

FAX機能の制御をソフト的に処理するCLASS1、ハード的に処理するCLASS2、それに両CLASSの弱点をカバーしたと言われるCLASS3？とFAXの規格も改変されてくるものと思われます。

したがって、現時点では両CLASSのFAX機能を搭載したFAXモデムを導入しておくほうが、

FAX通信ソフトとの対応の幅は広がるものと考えられます。

パソコンの周辺機器やソフトを求める場合には、使用している環境に適合したものを求めるのが最善と言われていますが、FAXモデムやFAX通信ソフトの場合も同様なことが言えますし、その点に十分留意して購入されることが肝心と言えましょう。

FAX通信ソフトは、DOS版かWindows版か

データモデムには専用の通信ソフトは、バンドルされていないのが普通ですが、外国産のモデムには付属している場合もあります。

特にFAXモデムの場合は、専用ソフトまたはそのモデムに対応したFAX通信ソフトが添付されているのが通例になっており、国産のFAXモデムもその例にならって対応ソフトが添付されています。

専用のFAX通信ソフトの場合は、当然のことながら添付されているFAXモデムでしか使えませんが、FAXモデムのほうは専用以外のFAX通信ソフトでも使えるように設計されています。

「汎用のFAX通信ソフトで、わが社のFAXモデムに対応しているものがあれば、開発費をかけて作らなくても済むのですが……」とメーカーサイドでは言っているものの、どこのFAXモデムにも対応できるFAX通信ソフトが無かったのも事実です。一方、メーカーサイドとしては、購入したその日から使用でき、しかも確実にFAX通信ができるものでないとメーカーの信頼を失いかねないとの理由もあって、完全対応のFAX通信ソフトをバンドルしているものと思われます。

そして、現在国産のFAXモデムにバンドルされているFAX通信ソフトはほとんどがDOS用で、特に98用ではWindows用のFAX通信ソフトはSTARFAX9624マルチパックだけです。国産では汎用でもWindows用のFAX通信ソフトは少ないのと、開発が遅れている関係上DOS用で対応させたほうが有利と見る向きもあります。98で動作するWindows用FAX通信ソフトの開発が遅れている最大の理由は、98用Windowsのシリアルの通信速度が9,600bpsまでしか対応していないという点にも起因しています。

それ故、FAX通信に必要な19,200bpsの通信速度が得られず、Windows 3.0Aや3.0Bの場合は、シリアルの通信速度（COMM.DRVの対応速度）を変更する方法を講じなければならず、NEC側では対応してくれないため、使いたいユーザーは通信のプログラムに明るい人が自作したフリーウェアの助けを借りて対応するほかありませんでした。

しかし、時代の要求に応じて98用Windowsのシリアル通信速度もMicrosoft Windows 3.1(98用)では19,200bpsまで対応できるようになり、Windows用FAX通信ソフトの開発に拍車がかかるものと思われます。

DOS用のFAX通信ソフトもWindows用のFAX通信ソフトも、プリントエミュレータ機能でFAX送信用のファイルを作る点に変わりはありませんが、アプリケーションソフトの印刷機能からFAXドライバを起動する場合に、アプリケーションソフトとの相性とかFEPやほかのドライバ類との割り込みなどの不具合の度合いを考慮に入れると、DOS用のFAX通信ソフトよりもWindows用のFAX通信ソフトのほうが有利なように思われます。

長期間使用したわけではありませんので、異論もあるものと思われますが、私が使用した感じではWindows用のアプリケーションソフト（ワープロや表計算ソフト）に送信原稿を読み込んで、加工したり編集したりして印刷機能でFAXドライバを起動して送信したほうが、DOS用のFAX通信ソフトで同じ作業をするときよりも手軽で快適に行えるように思われてなりません。

Windows自体の環境改善やWindows用のアプリケーションソフトの機能改善が日進月歩で行われる時代が到来したようですから、FAX通信ソフトも今後ますます使いやすくなっていくものと思われます。そして、国産のWindows用FAX通信ソフトも国民性や国内の通信事情を加味して、より簡便に、より使いやすくなっていくものと期待しています。

5

市販FAXモデムのテスト

谷沢 俊昌

データ通信を行うには通信ソフトが必要ですが，FAX通信の場合もFAX通信ソフトが必要になります。「FAXモデムは，バンドルされている専用ソフトでないと駄目だ」，「1頁だけなら送れるが，複数頁にわたると送れない」，「A君の所には送れるが，B君の所には送れない」，「ウィンドウズ用のFAX通信ソフトを買ったんだけど，DOS/V用なので，98ではFAXが送れない」等々，FAXモデムを使っている友達から，よくこんな話を聞きます。

パソコン雑誌やパソコン通信ネットのボードを賑わしているFAXモデム，元はと言えばデータ通信用のモデムとして開発されたものですが，FAXの普及に伴いデータ通信用のモデムに付加価値を与える意味でFAX機能を搭載したものです。

それ故，FAX通信専用のモデムは存在せず，データ通信用のモデムの拡張機能としてFAX機能がサポートされています。

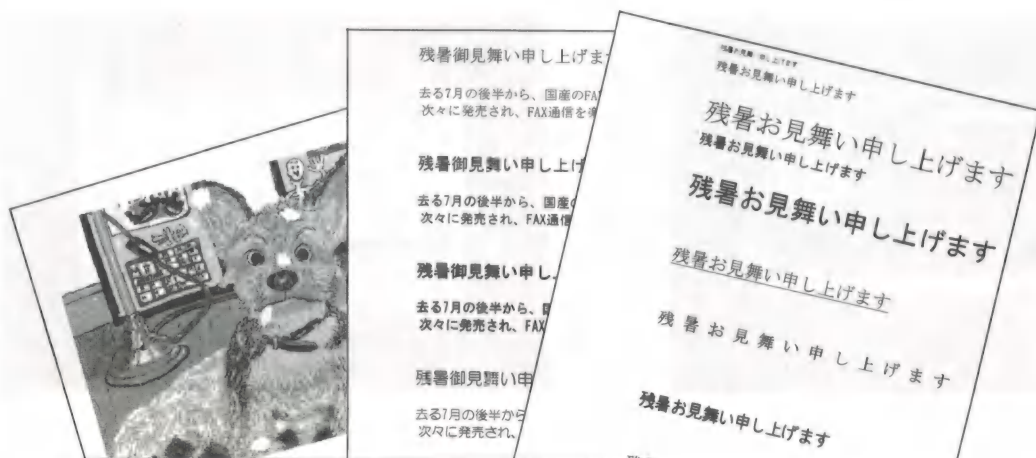
しかし，今回はFAXモデムの特集ですので，FAX機能だけにポイントを絞ってテストしました。

まず，テストに使用した環境，FAX通信ソフト，テストしたモデムの紹介といった順序で進めて行くことにしましょう。

パソコン，他の使用環境およびテスト方法

NECのPC-9801，DOS/V，MAC，他パソコンの機種も多く使われていますが，今回はPC-9801用に発売されているFAXモデムとFAX通信ソフトを取り上げました。使用したパソコンは9801DA/U2で，MS-DOSのバージョンは3.3C，ウィンドウズはNECの3.0A，およびマイクロソフトから発売されている98用の3.1を使用しました。

テスト用の原稿はワープロで作った3頁のものを使用して，テキストファイルだけの送信テストに留めました。また，NTTの電話回線を利用してテストを行うのが妥当ですが，テストのために沢山のFAXを送信しなくてはならないため，自宅内で家庭用電話交換機(ニューテク製NT-104)を使い，受信側は留守番電話兼FAX機，シャープ製のDA-7000を使用しました。

また，家庭内で行っていても1回目のテストではNGだったものが，2回目のテストではOKになるケースも生じたため，1つのテスト項目に対して3回のテストを重ねました。

【図1】送信例

使用したFAXモデムは1台を除いてテスト時点で市販されているものを使用しましたが，FAX通信ソフトの場合は開発中で機能制限付きのものもありましたので，受信テストについては今回は割愛することにしました。

いずれ，FAXモデムが普及し，FAX通信ソフトが完全な形でテストできるようになりましたら，受信機能のテストも交じえてレポートしてみようと思っています。送信例を図1に示しておきます。

テストに使用したFAX通信ソフト

DOS用の汎用FAX通信ソフト2本とWindows用の汎用FAX通信ソフト3本ですが，Windows用については，3本中2本がDOS/V用に開発されたものを使用しており，現時点では国産のWindows用の汎用FAX通信ソフトはEasyFaxのみとなっています。

①STARFAX V2.1(メガソフト)　¥30,000

メガソフトのFAXモデム用に開発されたDOS用のFAX通信ソフトで，同社のFAXモデムやエプソンのFAXモデムMX-240用に対応させて発売したもので，正式対応は歌っていないもののCLASS 1のFAXモデムでは使用できるものとして，評判の高いFAX通信ソフトです。

私の場合は，DOS用FAX通信ソフトのリファレンスとして愛用しており，新しいFAXモデムのテストには必ず使用しています。

プリンタエミュレータ機能，同報送信機能，時刻指定送信機能など，便利な機能も搭載されています。簡単な手順でFAXが送れ，しかも通信状況がビジュアルに確認できるのが特徴で，FAX通信が楽しくなります(**写真1**)。

FAX通信ソフトの老舗的な存在で，最近ではSTARFAX for Panasonic，STARFAX for AIWAといったOEM供給もしています。

②FAXPLUS(ランドマークテクノロジー)

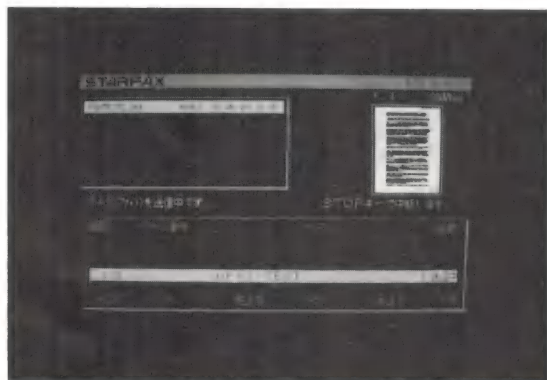
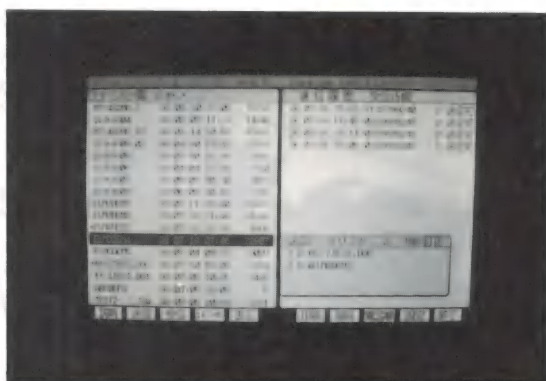
一時，アイワのPV-AF144V5に体験版として添付されていたもので，CLASS 1対応DOS用の汎用FAX通信ソフトで，製品版の発売期日，価格などは未定です。ファイル選択画面でドライブ切り替えができたり，FAX用ファイル作成の模様がウィンドウで表示されたり，ユニークな画面づくりがなされており，製品版の発売が楽しみと言えましょう(**写真2**)。

③EasyFax(エー・アイ・ソフト)　¥19,800

Windows用の汎用FAX通信ソフトとしては，初の国内制作版で，Windows人気の上昇にタイミングを合わせて発売されました。

ユニークなのはWindowsならではの電話帳機能で，ワープロやデータベースで培ったノウハウをFAX通信ソフトに取り込み，ビジュアルで便利な電話帳が作成できます。

また，Viewer機能で送受信内容のイメージをリアルタイムに画面に表示させたり，イメージを拡大／縮小／回転表示させることもできます。

ウィンドウズ上の各種アプリケーションで作成された文書や画像がFAX送信できるほかに，WindowsのDOS互換BOXから呼び出されるMS-DOSのアプリ

〈写真1〉 STARFAX V2.1

〈写真2〉 FAXPLUS

ケーションからも，その印刷結果をプリンタエミュレーション機能により送信することができます。CLASS 1 対応のFAX通信ソフトです(**写真3**)。

④日本語BitFax for Windows ¥19,800

Bit Software，IncがDOS/V用に開発したFAX通信ソフトで，威高テクニカル株式会社により日本語版が作成され，兼松セミコンダクター株式会社により輸入されたものです。

今回使用したのは，PanasonicのFAXモデムTO-700BにバンドルされたDOS/V版で，98Windows用ではありませんが，使ってみました。

CLASS 1，CLASS 2，SENDFAXにも対応しているFAX通信ソフトで，インストールの際に使用するモデムを認識させクラスを判別させなくても，使用時にソフト側がモデムのクラスを判別して，そのクラスに適合したクラス用のエンジンをロードする仕組みになっています。

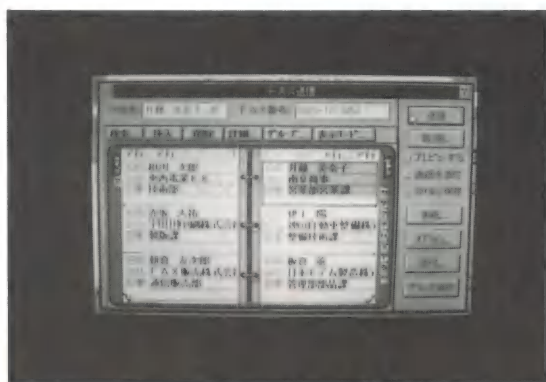
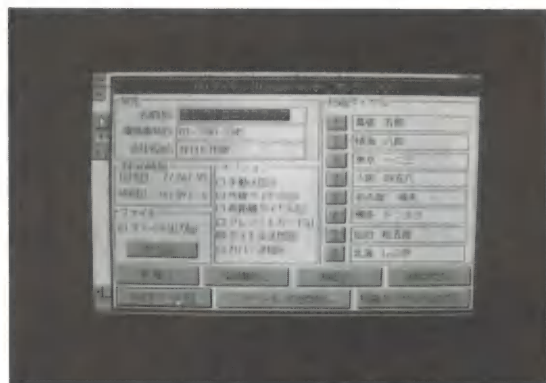
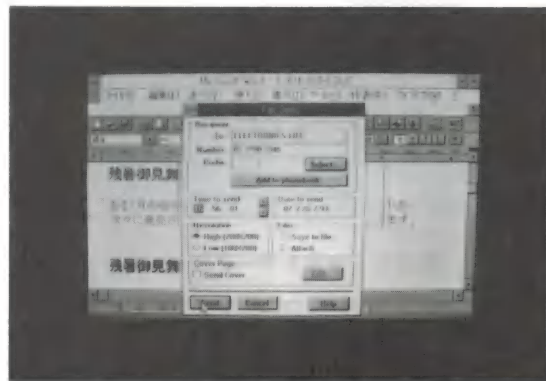
したがって，使用するモデムのクラスが不明でも使える利点もありますが，クラスがわかっていても使用者側では選択できない不便さもあります。つまり，CLASS 1，CLASS 2 両方を搭載している場合に，CLASS 1 で駄目だったらCLASS 2 に変えたいと思っても，それができません。FAXクラスの自動選択に加えて手動選択ができれば最高だと思われます。

接続時にCONNECTと表示されれば，CLASS 2 で接続されたことになります。CONNECT 9600/FAXと表示されれば，CLASS 1 で接続されたことになります。

アイコン操作で拡大／縮小／回転などの表示できるのも便利で，FAX通信ソフトとしてのひと通りの機能が搭載されています。

今回使用したのはDOS/V版ですが，98版もリリースされるそうです(**写真4**)。

⑤WinFax LITE(輸入品・英語版)

DELRINA社が開発したDOS/V用のWindows版FAX通信ソフトWinFax PROのOEM用バージョンで，カバーページほかを外した機能縮小版ですが，アメリカ製のFAXモデムにバンドルされていることが多いため，日本でも沢山のファンを持っています(**写真5**)。

〈写真3〉 Easy Fax

〈写真4〉 日本語BitFax for Windows

〈写真5〉 WinFax LITE

現時点では日本語版は発売されておりませんが，私も愛用しているソフトですので，今回はWindows用のリファレンスFAX通信ソフトとして新旧2つのバージョンを起用しました。

モデムチップの製造時期による違いや，プログラムのバージョンの違いによって，モデムとの相性の違いが認められます。

インストールの際にCOMM PORTとモデムのチェックを行い，自動認識で適合したクラスのプログラ

ムをロードする方式を採用していますが，認識できないモデム用に個別のパラメータも用意されており，モデム別の細かい設定も可能になっています。

OEM用の機能縮小版ですが，View機能や電話帳機能等Windows版のFAX通信ソフトとしての標準的な機能はすべて搭載しています。

今回テストしたFAXモデム

日本製8台，アメリカ製でJATEの認定を取ったもの2台，10台を選びました。

1 マイクロコア MC24FA5-P ￥24,800

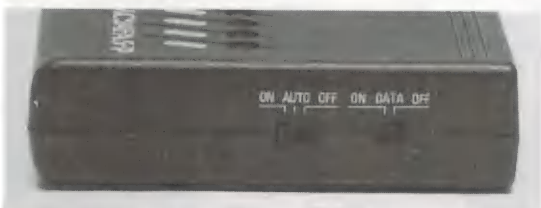
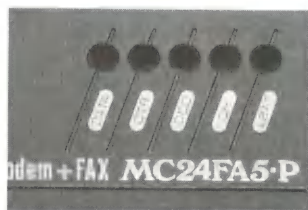
小型・軽量でハイコストパフォーマンスを実現している携帯型インテリジェントFAXモデムで(**写真6**)，FAX部はCLASS 1，CLASS 2をサポートしています。このFAXモデムには専用FAX通信ソフトLiteFAX98がバンドルされており，このソフトでもDOS上からFAX通信は可能ですが，テスト結果一覧でもわかるように，ほかの汎用FAX通信ソフトにも対応しています。

電源は単3乾電池4本，またはACアダプタ両方が使える仕様になっています。

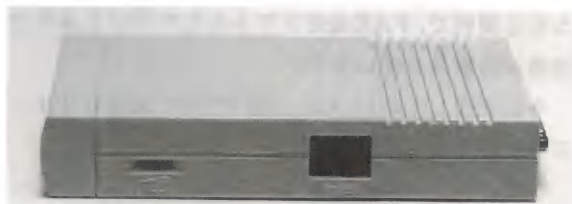
8MHz系でもFAXが送れるRockwellの新チップが搭載されており，20,800bpsの通信速度が得られるので，98の8MHz系のパソコンユーザーには喜ばれています。マイクロコアからは144/1414FAXモデムも発表されており，この雑誌が発売される頃には店頭を賑わせているものと思われます。

2 エプソン MX-240 ￥44,800

エプソン用として，STARFAX V2.1の対応FAX

〈写真6〉
MC24FA5-Pの外観とリアパネル

〈電源スイッチとデータスイッチ〉

〈LED表示〉

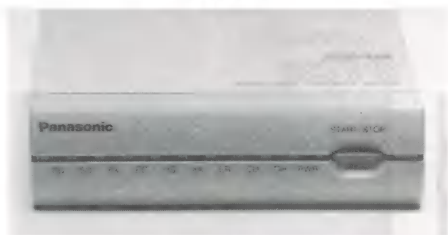
〈底面にあるバッテリ収納部〉

〈写真7〉
MX-240の外観

〈右側面にある音量調整器と電源スイッチ〉

〈フロントパネルにあるLED表示とデータスイッチ〉

〈リアパネル〉

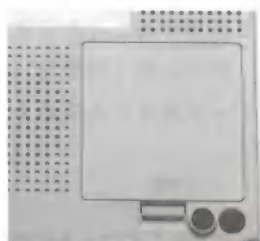
〈底面にあるディップスイッチ〉

〈写真 8〉
TO-700Bの外観

〈フロントパネルにあるLED表示とデータスイッチ〉

〈リアパネル〉

〈底面にあるディップスイッチ〉

〈写真 9〉
SMD-70の外観

〈底面にあるバックアップ
バッテリ収納部〉

〈リアパネル〉

モデムとして開発されたもので，98環境でも当然使用できます。CLASS 1 対応でWindows用FAX通信ソフトにも対応しています(**写真 7**)。

3 パナソニック TO-700B ¥44,800

パナソニックブランドで発売されているボックス型FAXモデムで，仕様も大きさもエプソンと同系ですが，FAX機能はCLASS 1 とCLASS 2 を搭載しています(**写真 8**)。

FAX通信ソフトはバンドルされておりませんが，98用にSTARFAX for Panasonic，Windows用にDOS/V版ですがBitFax for Windowsが用意されており，98版も発売されるようです。

テスト結果一覧でもわかるように，DOS版Windows版いろいろなFAX通信ソフトとの相性も良く，アイワとトップを争う結果が出ています。

4 ソニー SMD-50/70 ¥39,800/59,800

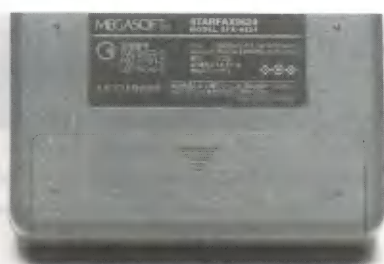
BBS機能を搭載したIT-V1200，2400を土台にFAX機能と着信時の自動切り替え機能を付加したインテリジェントモデムで(**写真 9**)，1 (4) MBのメモリーを内蔵してFAXのメモリー送信も可能にしています。FAX通信ソフトとしてはフリーウェアのFTRAN 3 が添付されていますが，SMD用にエトナ株式会社によって開発されたSMDtalk-98も発売されています。

このモデムにはCLASS 3 ？(SENDFAX系)が搭載されているので，CLASS 1，CLASS 2 用のFAX通

〈写真10〉 STARFAX9624の外観

〈リアパネル〉

〈左側面にある電源スイッチ〉

〈LED表示〉

〈底面にあるバッテリ収納部〉

信ソフトではテストできませんでしたが，WinFax LITEのSENDFAXコマンドでFAX送信ができることを確認しています。

SMDの特徴である内蔵メモリーを利用して，FAX送信用のファイルを別のSMD経由で任意の宛先にFAX送信するFAX転送機能は，新しいFAX送信の方法として注目されています。

いずれは，Windows上からFAX送信ができるSMD対応のFAX通信ソフトも開発されるものと期待しています。

5 メガソフト STARFAX 9624マルチパック ￥89,800

日本を代表するFAX通信ソフトの開発元にメガソフトのFAXモデムで(**写真10**)，98用に開発された9624に98用，DOS/V用のSTARFAX，さらにWindows用のFAXドライバーをセットしたマルチベンダー版。

このモデムは添付のFAX通信ソフトでないとFAX通信ができないので，汎用FAX通信ソフトでのテストは除外しました。

このモデムにセットされているSTARFAX 9624，およびSTARFAX for Windowsを使用すれば，通常のFAX通信のほか，モデムからダイレクトにFAX機にFAXを送ることもできますし，FAX機をスキャナ一代わりに使用することもできます。

これはたいへん便利な機能で，電話回線や家庭用の電話交換機を使わなくてもFAX機をプリンタとして活用することができます。

9624に続く新製品1414も発表されていますので，データ通信にも威力を発揮するものと期待しています。

6 オムロン MD96XL10V ￥46,800

マイルドセブン1個大のコンパクトサイズで，FAX/DATA通信を実現させたポケットモデム(**写真11**)で，単3乾電池とACアダプタの2電源方式が採用されています。

CLASS1対応で，FAX通信ソフトとして「OMRON FAXソフトウェア」がバンドルされています。このソフトはMD96XL10V専用ですので，ほかのFAXモデムには流用できませんが，モデムのほうはCLASS 1対応のFAX通信ソフトであれば，問題なく利用できます。

現在のところDOS用もWindows用も対応しているFAX通信ソフトが少ないようですが，14,400bps対応のMD144XT10Vも発表されており，FAX通信ソフト面での対応も広がって行くものと思われます。

7 インテグラン 通信ポコFAX MP96 ￥39,800

ラック型の業務用モデムを著名なインテグランの丸型で個性的なモデム「通信ポコ」のFAX機能搭載機で，CLASS 2だけに対応しています(**写真12**)。バンドルされているFAX通信ソフトは，日本語SUPER FAXのDOS/V版でしたが，98Windows上からも送信できました

今回テストに使用したFAX通信ソフトは，CLASS 2に対応しているものが少なかったため，芳しくない結果となりましたが，CLASS 2対応のソフトが開発されれば，対応するソフトも増えて来るものと思われます。

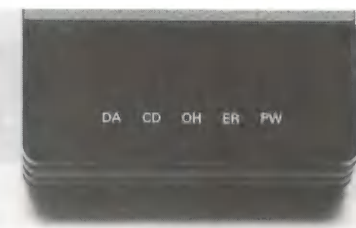
006P乾電池とACアダプタの2電源方式が採用されています。

8 アイワ PV-AF144V5 ￥64,800

国産初の14,400bpsFAXモデムとして発表されるや注目を集めた製品で，液晶表示機能，CLASS 1およびCLASS 2対応，8MHz系のパソコンからもFAXが送れるとあって，買えない人も多かったようです。発売当初はROMの不具合で悪評も飛び交いました

〈写真11〉 MD96XL10Vの外観

〈右側面にあるスイッチ類〉

〈リアパネル〉

〈LED表示〉

〈底面にあるバッテリ収納部〉

〈写真12〉
MP96の外観

〈左側面にあるRS-232C, モジュラージャック類〉

〈底面にあるバッテリ収納部〉

〈右側面にあるACアダプタ入力端子〉

が，製品も安定してきています。発売当初はFAX PLUSの体験版を添付しておりましたが，現在ではSTARFAX for AIWAが添付されています(**写真13**)。

日本製のFAX通信ソフト，アメリカ製のFAX通信ソフトともに広い対応性が発揮されており，対応の幅の広さはテスト機中トップでした。

FAXも14,400bpsに対応していますが，14,400bpsで通信できるFAX機がなかったので，9,600bpsでのテストに留めました。

機会があれば，ぜひ14,400bpsでのFAX通信も味わってみたいと思います。

9 MultiTech MT1432BAI ￥128,000

アメリカの業務用通信機器メーカーMultiTech のMultiModemIIシリーズの製品で，日本ではオール・テクノロジー・グループ株式会社が輸入してJATEの認定を受け，日本語版マニュアル，通信ソフト，FAX通信ソフトをバンドルして発売しています(**写真14**)。

CLASS 2 に対応しており，専用ソフトMulti Express FAXの98版もありますが，今回はWindows用のFAX通信ソフトでテストしてみました。

CLASS 2 対応のソフトでは，すべて完璧なFAX通

〈写真13〉 PV-AF144V5の外観

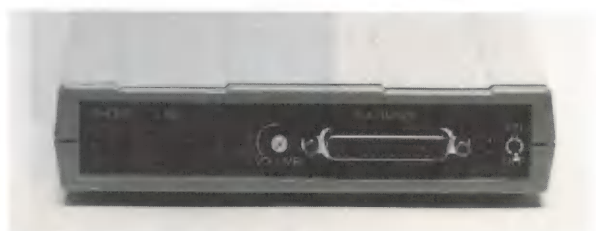
〈右側面にある音量調整器とディップスイッチ〉

〈フロントパネルにある液晶表示とスイッチ類〉

〈リアパネルと電源コードに付いているノイズフィルタ〉

〈写真14〉 MT1432BAIの外観

〈右側面にあるディップスイッチ(左)，専用線切り替えスイッチ〉

〈フロントパネルにあるLED表示，データスイッチ〉

〈リアパネル〉

〈写真15〉 Scout PLUSの外観

〈フロントパネルにあるLED表示〉

〈リアパネル〉

信が確認できました。

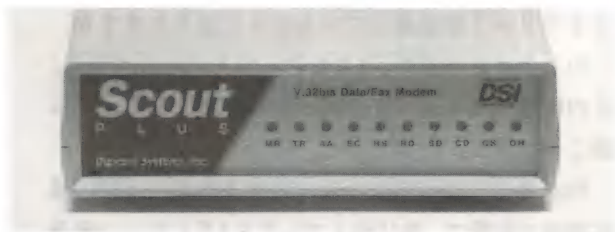
10 Digicom Scout PLUS ¥118,000

ラック型モデムの製造も手掛けているアメリカの通信機器メーカーDigicom社のFAXモデムで，日本では岩谷産業が輸入して株式会社シーアンドピーから発売されている14,400／14,400FAXモデムです(**写真15**)。JATEの認定を受け，日本語マニュアル，WinFax LITE(英語版)をバンドルして提供しています。

CLASS 1対応で，WinFax LITEや国産のDOS用FAX通信ソフトとは相性が良いのですが，国産のWindows用のFAX通信ソフトとは相性が悪いようです。

テスト結果

各社のFAXモデム／FAX通信ソフト別テスト結果を**表1**に示します。〇印は3頁の原稿を3頁完全に送信できたものです。

△印は3頁の原稿中1頁しか送信することができなかったもの，つまり頁切り替えがうまくゆかず，1頁送信したところでFAX機が通信エラーを起こして止まってしまった場合で，1頁の原稿に差し替えて1頁なら完全に送れることを確認したものに付けてあります。

×印は「モデムからの応答が得られず，接続さえもできなかった」「接続はできたものの，頁の途中で送信が止まってしまい，FAX機側では通信エラーを起こしてしまった」「呼び出しができずタイムアウトになってしまった」などです。

テスト方法のところで1つの項目に判定を下すのに同じテストを3回ずつ行ったと書きましたが，同室内でしかも近距離でテストしていても，3回とも同じ結果が出ない場合もあるほどデリケートな状態に陥る場合もありました。

今回のテストは，パソコンも1台，FAX機も自宅で接続できなかった場合を除いては1台のFAX機に接続していますので，テストしたFAXモデムで常時すべて同様な結果が出るとは思われません。

私がテストした時には結果が悪くても，この雑誌が発売された時点では改良されているかも知れませんし，あくまでも参考資料として受けとっていただければと考えています。

FAXモデムのテストを終えて

短い期間ではありましたが，現在日本で手に入るFAXモデムとFAX通信ソフトに触れることができました。FAXモデムはアメリカで開発されただけあって，モデム本体にしてもFAX通信ソフトにしても日

本より２～３年は進んでいるように思われますが，このところ日本でのFAXモデムの人気からすると，その距離は年を追うごとに狭まって来ているように感じられます。

特に日本で50％以上のシェアを占めている98の通信環境の影響で，伸び悩んでいたFAXモデムの普及が今年後半には進むといった明るい見通しも期待できそうです。

それにしても，相手のFAX機に接続できたからといって送信が終わって相手から受け取ったという返事が来るまで安心できない不安な気持ちからは，一日も早く解放されたいものです。

FAXモデムでFAXを送った場合だけでなく，FAX機どうしでも製造メーカーや製造時期の違いで通信不良を起こすこともありますし，電話回線の状態によっても不具合を起こすことがあります。

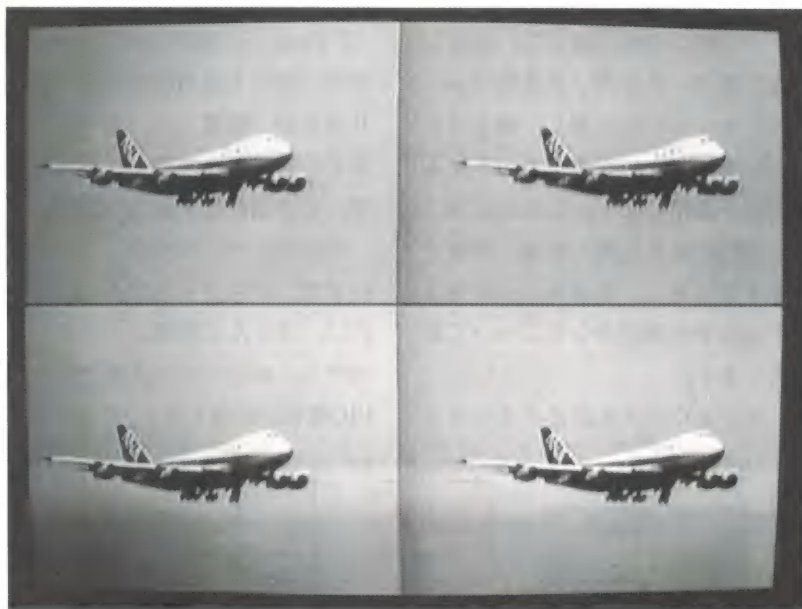
モデムメーカーやソフトハウスの努力で，ビクビクしながらFAXを送る状態を一日も早く解消して欲しいと思っています。

DOS用のFAX通信ソフトも結構ですが，98環境で使用する場合を考慮すると，ハード的な制約にとらわれ難いWindows用のFAX通信ソフトのほうが良いように思われ，FAXのクラスもCLASS 2にも対応したほうが良い結果を産むような気がしてなりません。

	メーカー・モデム機種	STARFAX CLASS 1	FAXPLUS CLASS 1	EasyFax CLASS 1	日本語BitFax CLASS 1,2自動	WinFax(新)		WinFax(旧)	
						CLASS 1	CLASS 2	CLASS 1	CLASS 2
1	マイクロコア MC24FA5-P	○	○	○	×	△	×	○	△
2	エプソン MX-240	○	×	○	○	×	none	×	none
3	パナソニック TO-700B	○	○	○	△	×	×	○	○
4	ソニー SMD-50/70	SMDtalk98, FTRAN3, WinFax SENDFAXで送信可能							
5	メガソフト SFA-9624	専用ソフトSTARFAX 9624以外は動作せず							
6	オムロン MD96XL10V	○	×	○	×	○	none	×	none
7	インテグラン 通信ポコFAX MP96	none	none	none	×	none	×	none	○
8	アイワ PV-AF144V5	○	○	△	△	○	○	△	△
9	MultiTech MT1432BAI	none	none	none	○	none	○	none	○
10	Digicom Scout PLUS	○	○	×	×	×	none	○	none

【表１】各社FAXモデム/FAX通信ソフト別テスト結果一覧

特集2

パソコンにTV画面が取り込める
フレームメモリーを使用した
多機能画像入出力システムの製作
最終編

企画・構成/解説：谷 和彦/土屋 徹

6月号・7月号と2カ月にわたって，多機能フレームメモリーの製作記を書いてきました。今月は，その最終回としてソフトウェア編をお届けします。プログラミング言語はC言語を使用して，フレームメモリーのRAMからパソコンのメモリーへの転送ソフト，パソコンのCRT画面への表示ソフト，パソコンのメモリーからフレームメモリーのRAMへの転送ソフトの3つのソースプログラムについて説明します。また，ビデオのカラー画面をパソコンの16色表示画面にできるだけきれいに表示するためのプログラムや，拡大表示・縮小表示プログラムについてはフローチャートを示します。さらに，ディジタル画像処理の基本となる2次元フィルタについてもお話します。

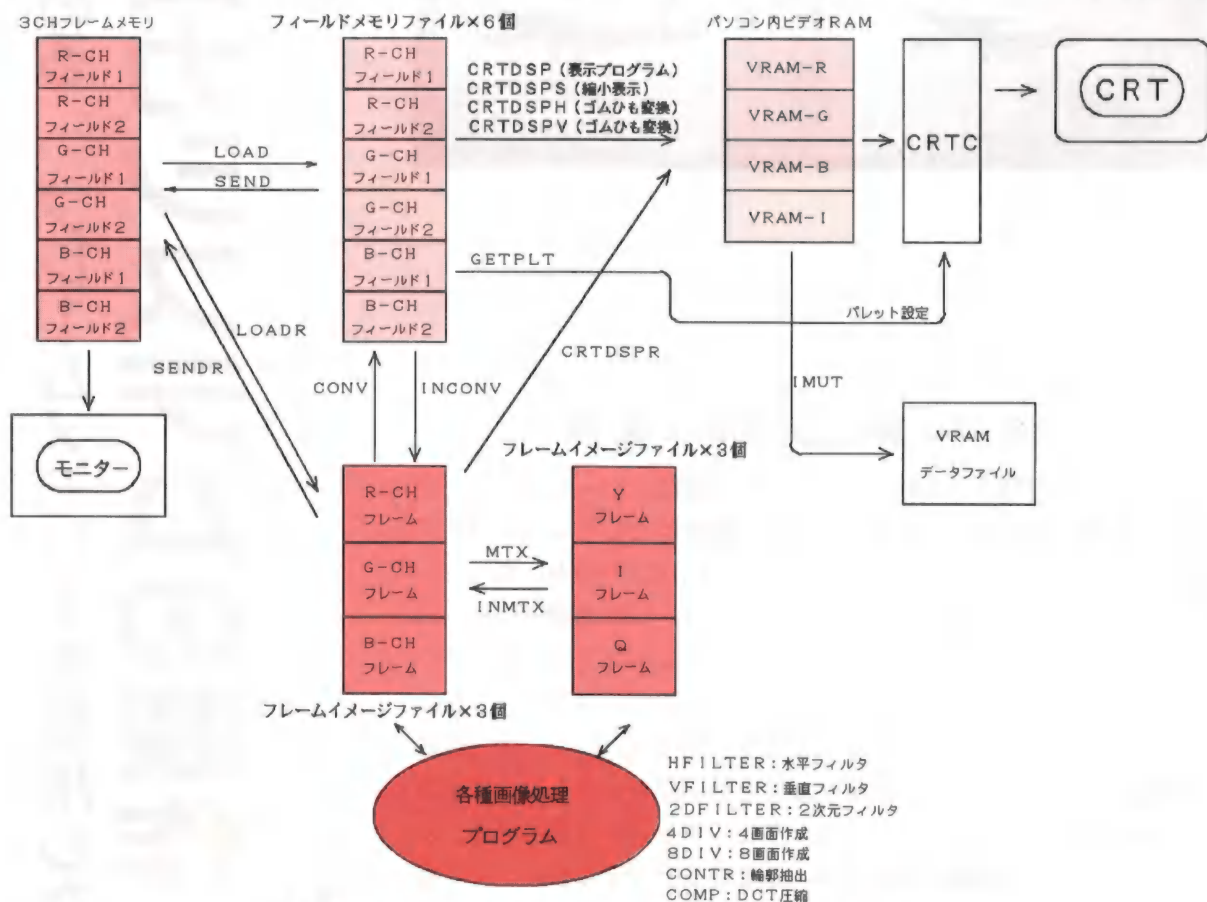
各種プログラムの解説

6月号・7月号で，多機能フレームメモリーのハードウェア製作を終了し，いくつかの表示例を示しました。

今回は,「多機能フレームメモリーの製作」の3回目として，基本的な画像入出力プログラムや，拡大縮小・輪郭抽出プログラムなどソフトウェアを中心に解説します。

前回は，パソコンに取り込んだ画像データをCRTディスプレイに表示する方法のみを説明しましたが，今回はこれに加えて，処理した画像データを再び多機能フレームメモリーに送り返し，通常のテレビモニタ上に画像を表示させる方法も説明します。これにより，本格的な画像処理の世界を体験できます。また，最後にハードウェアの若干の変更点などについて説明します。

ソフトウェアを記述するプログラム言語はマイクロソフト社のMS-C Ver5.1を使用します。BASIC言語の中にも使用できるものはありますが，最近では，筆者はほとんどC言語しか使用していないので，C言語に話を限定します。

使用するパソコンはPC-9801シリーズで，デスクトップタイプであれば，ほとんど問題ありません。ただし，48ビットの入出力を持つPIO基板が拡張スロットに装着でき

【図1】ユーティリティプログラムの基本構成と画像ファイル

なければなりません。PIO基板は前回も説明しましたが，48ビットの入出力を持つものであれば，安価なものでかまいません。筆者が使用したPIO基板はアドテックシステムのAB98-04Aという製品です。

1フレームのRGBデータは約1.4Mバイトもあるので，ハードディスクは不可欠です。なお，RAMディスクがあれば高速でアクセスできるので，非常に便利です。

後で説明しますが，工夫すればもう少し小さい容量のファイルに変換できます。

パソコンのディスプレイ機能については，4096色中16表示が可能な640×400ドットの高解像度タイプが最も一般的なので，以下の話はこのタイプの表示形式に話を限定します。最近，普及してきた256色表示や，640×480ドット構成の表示形式についてもお話ししたかったのですが，紙面の関係で省略します。

OSについて

今回使用したOS(オペレーティングシステム)は，MS-DOS Ver3.3Cです。また，MS-DOS Ver5.0でも基本的には問題はないようです。

MS-DOSでは，通常の使用法では，640Kバイトのメモリー空間しか使用できません。さらにDOSのシステムや，各種ドライバが常駐しているため，実際に使用できるメモリー空間はさらに狭くなっています。

最近のパソコンではXMSやEMSなどの拡張メモリーにより数メガバイトに及ぶメモリー空間を使用できるようになっていますが，あまり一般的ではないようです。

ところで，一般ユーザーがあまり苦労せず1Mバイト以上のメモリー空間を利用できる，DOSエクステンダという便利なソフトウェアがあります。これを利用すると非常にスマートなプログラムが書けるのですが，残念ながら使用できるC言語のコンパイラが限られているようです。

したがって，今回解説するプログラムはすべて，640Kバイトのメモリー空間で動作するように工夫しました。しかし、将来はDOSエクステンダを使用して，もっとスマートなプログラムにしたいと思っています。

それでは個々のプログラムを説明する前に，基本プログラムの全体構成と中間ファイルについて説明します。

ユーティリティプログラムの構成

図1にユーティリティプログラムの基本構成と，画像ファイルのタイプについて示します。また画像ファイルについては，以下に示すような3つのタイプとしました。

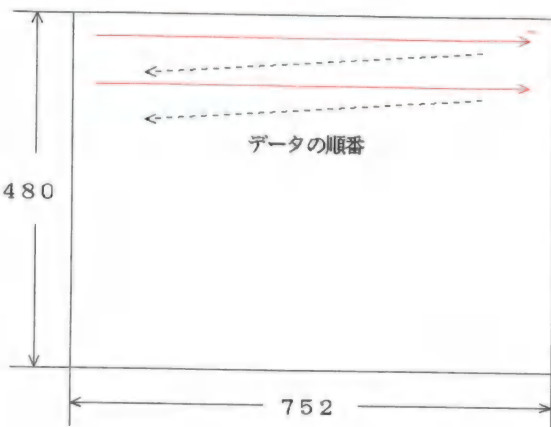
- タイプI(238875BYTE×6)6つのフィールドメモリーからのデータをそのままバイナリファイルに変換したもの。

- タイプII(360960BYTE×3)タイプIの2つのファイルから，有効データ期間のみを取り出し，フレームイメージに変換したバイナリファイル。

- タイプIII(ファイルサイズは切り取る領域の大きさで変化)VRAMに転送されたデータをイージカッタで切り取り保存するファイル。

またタイプIIのファイルには，RGB形式のものとYIQ形式の2種類があります。データの入出力はRGB形式ですが，各種画像処理には，一般的にYIQ形式のほうが好都合です。

タイプIIIのファイルを作成するソフトウェアでよく知られているものに，ジャストシステムから発売されている「IMUT．COM」(イメージカッタ)というプログラムがあり，筆者もよく使用しています。

【図2】フレームイメージファイルの形式

入出力プログラム

入力プログラムには，6個のフィールドメモリーからのデータを，そのままの形式で6個のバイナリファイル(タイプⅠ)に変換する「LOAD」というプログラムと，図2に示すフレームイメージに変換して3個のバイナリファイル(タイプⅡ)に変換する，「LOADR」というプログラムを作成しました。

リスト1にプログラムLOADのソースリストを示します。ここで，関数DATAGET(　)は1つのフィールドメモリーからデータを読み込む機能を持っています。

図3にフィールドメモリーよりデータを取得するプログラムの基本的な流れを示します。

プログラムLOADRは，単にデータをフレームイメージに変換するだけではなく，水平・垂直のブランキング期間を除いた有効画面領域のデータのみをファイルに変換するので，ファイルサイズが約76%に減少するため，ディスクを節約することができます。また，このファイル形式ならば，2HDのフロッピーディスク1枚にRGB1フレームのデータを記録することができます。

次に出力プログラムについて説明します。出力プログラムにも，タイプⅠのファイルを読み込んでフィールドメモリーに転送する「SEND」というプログラムと，タイプⅡのファイルを読み込んでフィールドメモリーに転送する「SENDR」というプログラムの2つがあります。

ここではプログラムSENDについて説明します。**リスト2**にプログラムSENDのソースリストを示します。

図4に，フィールドメモリーにデータを転送するプログラムの基本的な流れを示します。

ファイル変換プログラム

前に説明したように，ファイルタイプには3つのタイプがあり，さらに有効データの精度に劣化がないタイプⅠとタイプⅡは，相互に変換することが可能です。

【図3】1つのフィールドメモリよりデータを取得するプログラムの基本的な流れ

この目的のために「CONV」，「INCONV」という2つのプログラムを作成しました。

プログラムCONVはタイプIからタイプIIへの変換であり，プログラムINCONVはタイプIIからタイプIへの変換を行います。

16色カラー表示と白黒16階調表示

PC-9801とその互換機種の大部分は，4096色中16色表示が標準的な表示モードとなっています。4096は12ビットですからRGBそれぞれ4ビット(16階調)で割り当てられていますが，同時に発色できるのは16色すなわち4ビット分しかないので，RGBそれぞれ1.3ビットの割り当てしかないことになります。

したがってカラーモードを標準パレットで実現すると，原画とは似ても似つかないような(特殊効果のような)表示になってしまいます。しかし白黒モード(RGBを同じ値にする)にすれば，4ビット(16階調)が割り当てられるので，ある程度の表示ができます。筆者が実際に試したところ，白黒写真に近いものが得られました。もっとも，ビット数が少ないので擬似輪郭が生じます。

16色カラー表示と最適パレット設定プログラム

前に説明したように，標準パレット設定ではカラー表示を行うと原画とはかなり異なってしまいます。そこで，一枚の静止画中に存在する代表的な色(輝度を含む)を

【図4】
1つのフィールドメモリにデータを転送するプログラムの基本的な流れ

16色選択し，その色を使用してパレット設定する方法が考えられます。そしてそれぞれの画素に，一番近い色のパレット番号を割り当てて表示することで，原画により近い表示を得ることができます。

このような処理は，1枚の静止画中に存在するRGBの値の分布を統計的に分析し，16色のパレットを設定することで理想的に行うことができます(ベクトル量子化において，トレーニングシーケンスによりコードブックを作成する過程と基本的には同じです)。

START
ファイル名の入力
パラメータ設定
ファイル読み込み
白黒16階調モードに設定
ディスプレイに表示
パレット番号 1～16
カーソルを移動する
カーソルを固定して座標を記録する
ファイル読み込み
設定したポイントのRGBデータを取得する
16ポイントのパレットデータをファイル出力する
END

実際には、ファイルサイズが大きすぎて主記憶に一度に全部入らないので、分割してファイルをアクセスし、処理するようにしている。

【図5】プログラムGETPLTの処理の流れ

しかし，人間が絵を見て代表的な16色を選択しても意外とうまくゆくものです。また，画素にパレット番号を振り分ける過程において，パレットの16色と，各画素の色(輝度を含む)との差ベクトルの大きさを比較しますが，この差ベクトルの大きさはRGBそれぞれ同じスケールで定義して計算を行いました。

より良い表示を行うためには，人間の視覚特性を考慮する必要があり，RGB空間ではなく別の座標系に変換して比較すべきなのですが，話がむずかしくなりすぎるので，今回は簡単化してプログラムを作成しました。

さて，1枚の静止画に存在する代表的な16色を選択しパレットデータのファイルを作成するプログラム「GETPLT」の処理の流れを**図5**に示します。

このプログラムは，取得したデータを白黒16階調モードでCRTに表示し，ディスプレイ上に十文字形のカーソルを出して，キーボードでカーソルを移動して代表点を選択する形式になっています(**写真1**参照)。

したがって，テレビモニタに静止画を表示して色を確認しながら操作する必要があります。RGBそれぞれ1フレーム分のデータは，640Kバイトのメインメモリーに入りきらないので，プログラムを工夫して，ファイルからデータを読み込んでCRTに表示し，代表点の水平垂直座標を決定してから，もう一度ファイルからデータを読み込みパレットデータを取得しています。

この方法でCRTにカラー表示す

ると，テレビ画面を油絵の具で描いたようになります。

標準表示プログラム

次に，フィールドメモリーから取得したデータをCRTディスプレイ上に標準サイズで表示するプログラム「CRTDSP」を紹介します。このプログラムは，タイプIのファイルを入力しCRTディスプレイ上に表示するプログラムであり，白黒16階調表示と16色カラー表示を選択できるようになっています。

また，16色カラー表示のモードでは，プログラムGETPLTで作成したパレットデータファイルを入力し，最適な表示を行うことができます。標準サイズではCRTディスプレイの画面よりテレビ画面のほうがサイズが大きい(ドット数が多い)ので全画面を表示することができません。したがって，CRTディスプレイに表示するテレビ画面の水平・垂直始点を指定できるようにしています。

リスト3にプログラムCRTDSPのソースリストを示します。

標準表示プログラムには，タイプIIのファイルを入力して表示するプログラム「CRTDSPR」があります(説明省略)。

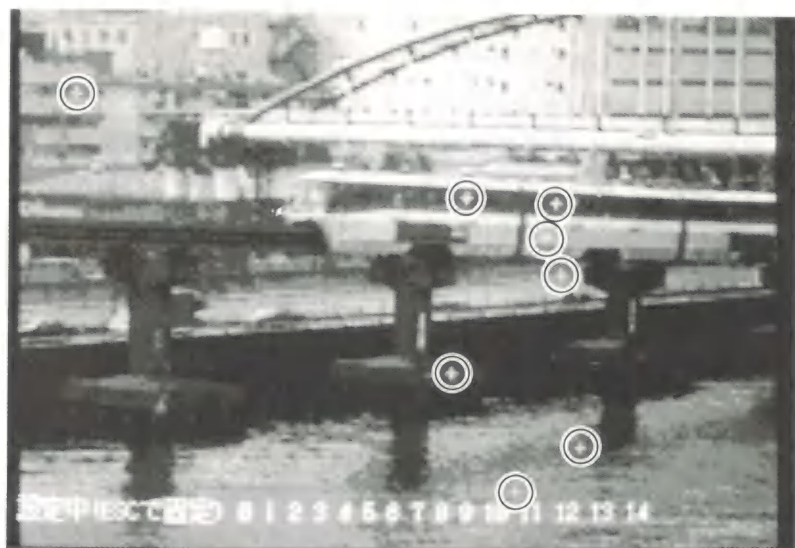



写真2に白黒16階調表示を，**写真3**に標準カラーパレットを使用した16色カラー表示を示します。

さらに**写真4**に，最適カラーパレットを使用した場合の表示例を示します。カラー印刷でないのが残念です。

〈写真1〉 GETPLTの画面(○印がカーソル)

拡大縮小表示プログラム

〈写真2〉
タイプ2ファイル
白黒16階調表示

〈写真3〉 タイプ2ファイル　標準カラーパレット16色表示

〈写真4〉 タイプ2ファイル　標準カラーパレット16色表示

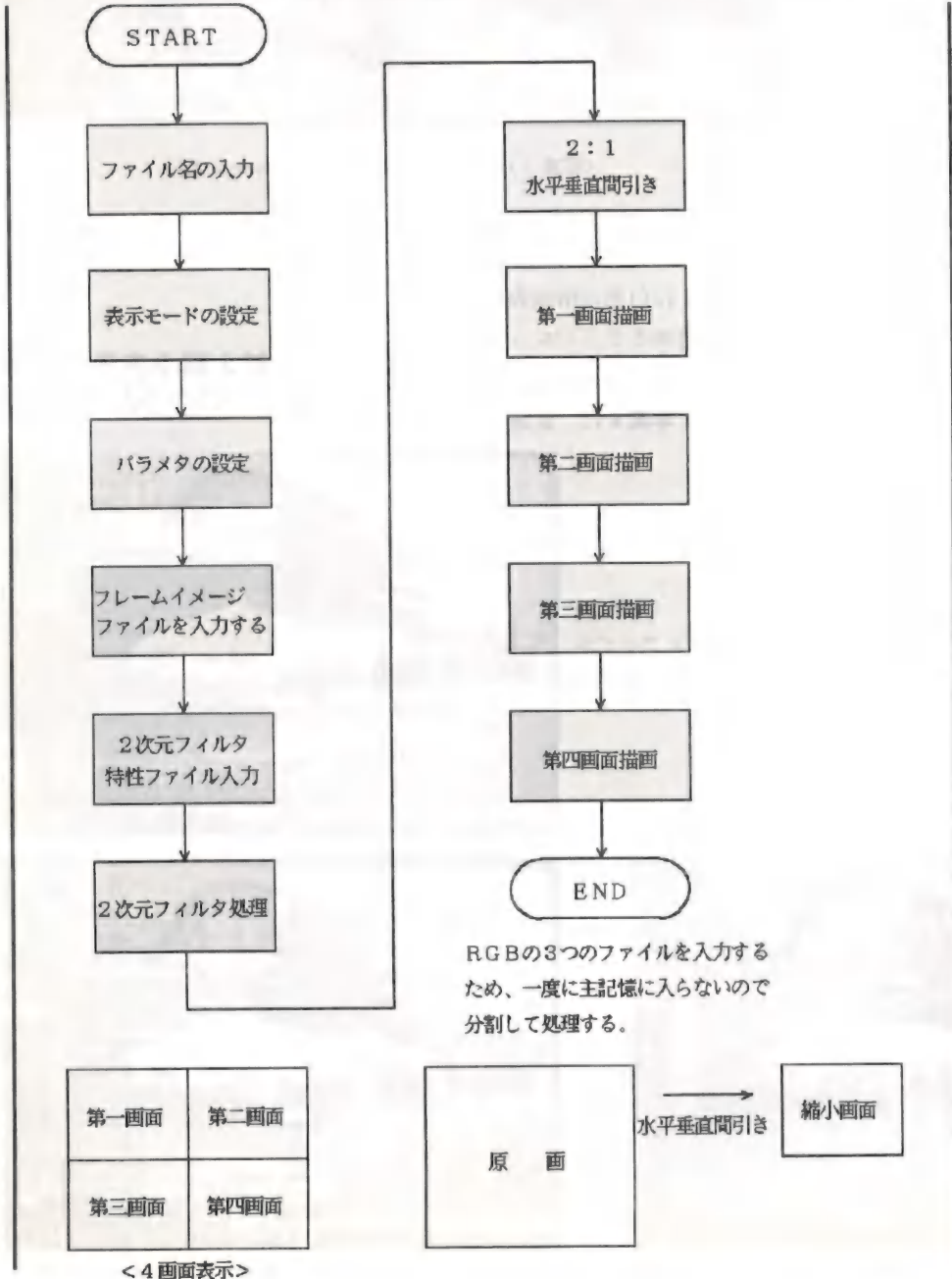
拡大縮小表示については，前回ある程度お話ししましたので，ここでは簡単に説明します。

最初に縮小表示について説明します。**図6**は，縮小表示プログラム「CRTDSPS」の基本的な流れを示します。拡大縮小表示プログラムは，タイプIIのファイルを入力するほうが話が簡単です。

プログラムCRTDSPSは，水平・垂直ともに1/2倍とした4画面表示をCRTディスプレイ上で実現します。このプログラムで使用した2次元プリフィルタの特性を**図7**に示します。**写真5**に縮小表示の例を示します。

ただし通常の自然画であれば，1/2倍程度の縮小率の場合にはプリフィルタ処理を省略してしまっても，さほど折り返しひずみは目立ちません。1/4・1/8倍については，読者のみなさんで考えてください。

1/4倍表示で使用するプリフィルタのタップ係数と，フィルタ特性を**図8**に示します。

次は，拡大表示について説明します。**図9**に拡大表示プログラム「CRTDSPL」の基本的な流れを示します。前回，拡大表示を行う場合に，同じ画素を複数並べて実現

【図6】
縮小表示プログラムの処理の流れ

するとブロックひずみが生じることをお話しました。そして，このひずみはフィルタ処理で除去できることもお話しました。しかし，フィルタ処理を使用して拡大表示を行う場合，同じ画素を複数並べる方法よりも，**図10**に示すようにゼロ内挿によって表示画素数を増やし，その後，２次元補間フィルタ処理でゼロ内挿点に値を生じさせる方法のほうがフィルタの設計

〈写真５〉
縮小表示例

〈 ２次元周波数特性〉

この２次元フィルタ特性は水平周波数と垂直周波数で変数分離できるため、
同じタップ係数の水平フィルタと垂直フィルタを縦続接続することで得られる

タップ係数　　　［垂直フィルタ：３１タップ，水平フィルタ：３１タップ］

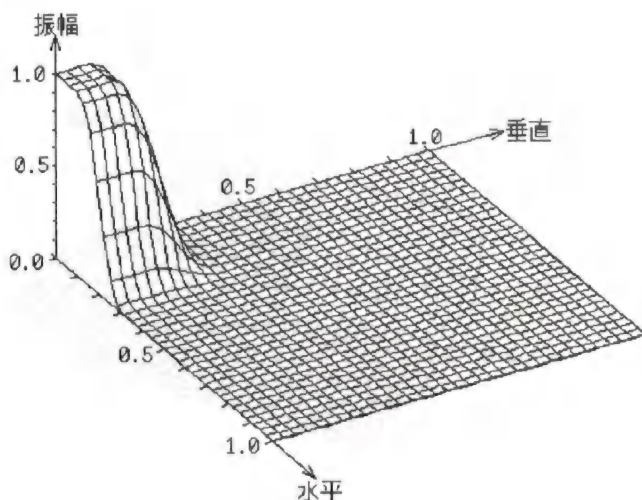
水平タップ番号	タップ係数	水平タップ番号	タップ係数
16	0.4484971016	8,24	-0.0215766796
15,17	0.3111540776	7,25	0.0022910309
14,18	0.0482025056	6,26	0.0126617534
13,19	-0.0871783396	5,27	0.0016887067
12,20	-0.0413377455	4,28	-0.0061474890
11,21	0.0358675822	3,29	-0.0021422958
10,22	0.0317604045	2,30	0.0022555776
9,23	-0.0129896478	1,31	0.0012420080

【図７】
水平垂直1/2の
縮小表示に使用する
２次元プリフィルタの特性

が容易です。

このような目的で使用するフィルタは，総じて補間フィルタと呼ばれます。**図11**に，2倍表示に使用する補間フィルタのタップ係数とフィルタ特性を示します。**写真6**に拡大表示の例を示します。

テレビ画面上への表示

さて，前節で説明した各プログ

〈２次元周波数特性〉

この２次元フィルタ特性は水平周波数と垂直周波数で変数分離できるため、
同じタップ係数の水平フィルタと垂直フィルタを縦続接続することで得られる。

タップ係数　　　［垂直フィルタ：３１タップ，水平フィルタ：３１タップ］

水平タップ番号	タップ係数	水平タップ番号	タップ係数
16	0.1987148318	8,24	-0.0215567318
15,17	0.1844470959	7,25	-0.0102983706
14,18	0.1457761715	6,26	-0.0003105412
13,19	0.0935411201	5,27	0.0052145566
12,20	0.0412995286	4,28	0.0061418056
11,21	0.0006296656	3,29	0.0042631549
10,22	-0.0223475210	2,30	0.0017586427
9,23	-0.0279783311	1,31	0.0000623385

【図８】
水平垂直1/4の
縮小表示に使用する
２次元プリフィルタの特性

ラムはCRTディスプレイ上への表示を行いますが，処理したデータをもう一度タイプIIのファイルに変換し，フィールドメモリー上に転送すれば，テレビ画面上に16777216色のフルカラー表示をすることができます。

取得した画像データに各種画像処理を行い，その結果をテレビ画面上に表示する場合の処理の流れと，使用するプログラムについて**図12**に示します。

各種画像処理は，フレームイメージに変換して行います。拡大縮小表示のために「4DIV」(縮小4画面表示)，「ZOOM」(拡大表示)という2つのプログラムを作成しました。

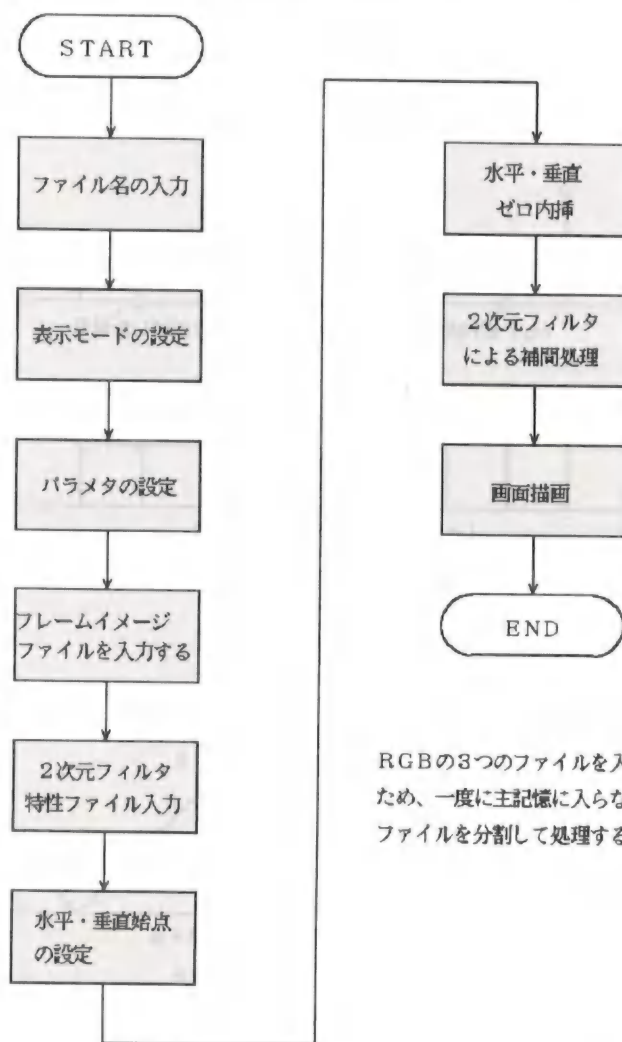
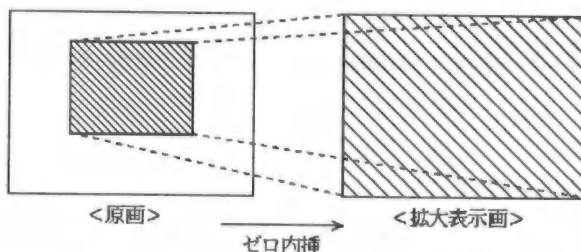
プログラムの説明は省略しますが，**写真7**に縮小4画面表示，**写真8**に2倍拡大表示の例を示し，**写真9**にその原画を示します。写真5・6と比較してみてください。

フィルタ処理と輪郭抽出

前に少しフィルタ処理のお話をしましたが，ここではもう少し詳しく説明し，輪郭抽出を応用例として示します。

画像処理では，非常にさまざまなフィルタを使用します。フィルタ処理といっても，広い分野の話になってしまうので，ここでは線形のFIR(有限長インパルス応答)フィルタにより，周波数軸上での信号成分の分離に話を限定します。

画像処理では，1次元フィルタだけではなく2次元フィルタ処理や3次元フィルタ処理もよく使用されます。特に動画の処理では，3次元フィルタ処理が色々な目的で使用されています。しかし，リアルタイム処理を行う機器で3次元フィルタを構成すると，どうしても時間軸方向(テンポラル方向とも言う)のタップ長が長く取れません。

それは，3次元フィルタにおいて時間方向の単位遅延素子はフレームメモリーやフィールドメモリーになるため，タップ長を大きく取ると大容量のメモリーが必要となるからです。また，垂直方向の単位遅延素子はラインメモリーであり，水平方向の単位遅延素子はDフリップフロップということになります。したがって，2次元フィ

【図9】拡大表示プログラムCRTDSPLの処理の流れ

ルタ処理までなら比較的タップ数を大きくできます。また，静止画では２次元フィルタ処理までを考慮すれば十分です(時間軸方向に変化がないため)。

　さて２次元フィルタ処理は，フィルタ特性を反映している２次元インパルス応答(図13参照)と式(1)に示す２次元の畳み込み演算(コンボリューション)を行うことで実現できます。

$$y(n_1,\ n_2)=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}x(k_1,\ k_2)\cdot h(n_1-k_1,\ n_2-k_2)\cdots\cdots(1)$$

$x(n_1, n_2)$：入力数列

$y(n_1, n_2)$：出力数列

$h(n_1, n_2)$：２次元インパルス応答

振幅

時間

(a) 原信号をサンプリングした信号

振幅

時間

(b) ゼロ内挿により時間軸を２倍に引き延ばした信号

振幅

時間

(c) 補間フィルタ処理後の信号

振幅

周波数

(A) 原信号のスペクトル

振幅

Fs　2Fs　3Fs

周波数

(B) サンプリング後のスペクトル

振幅

Fs　2Fs　3Fs

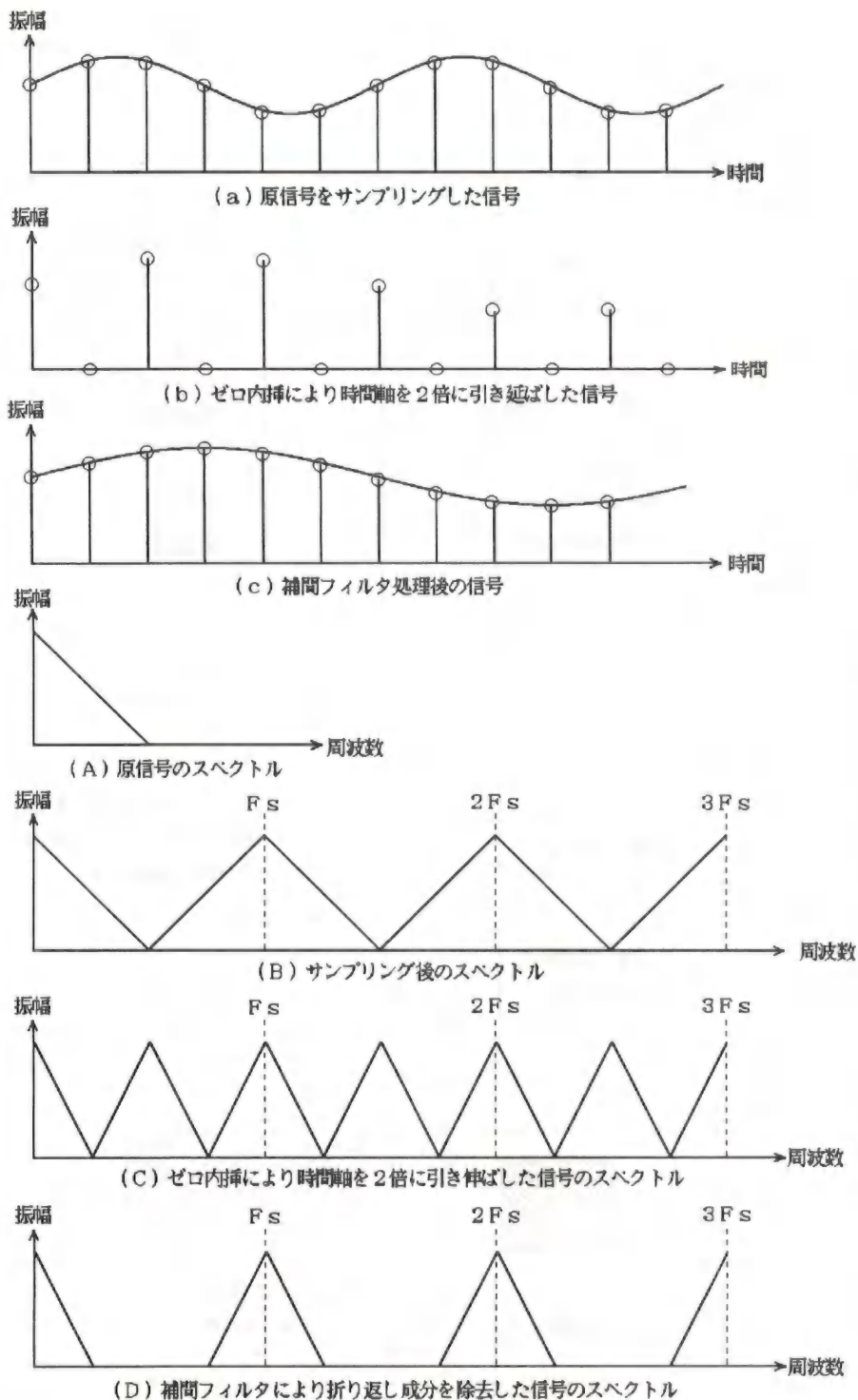
周波数

(C) ゼロ内挿により時間軸を２倍に引き伸ばした信号のスペクトル

振幅

Fs　2Fs　3Fs

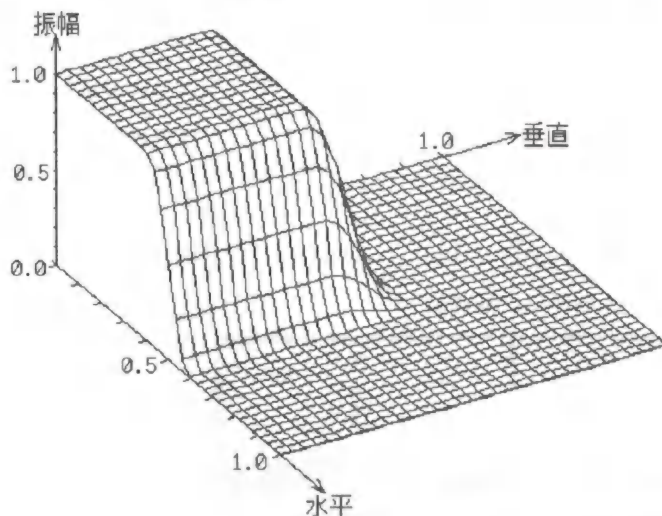
周波数

(D) 補間フィルタにより折り返し成分を除去した信号のスペクトル

【図10】

ゼロ内挿＋補間フィルタ処理による２倍拡大表示

図14・15・16にフィルタ特性とタップ係数の例を示します。また，**写真10**に示すゾーンプレート信号に対して，それぞれの特性のフィルタ処理を行った結果を**写真11・12・13**に示します。

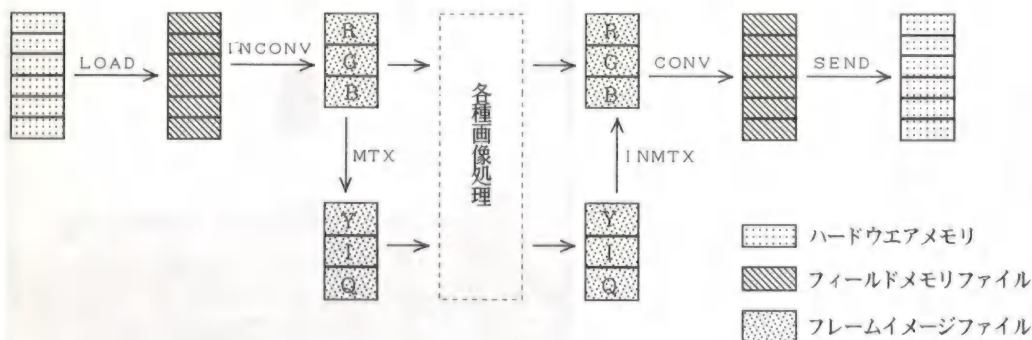
さて，フィルタ処理の応用例として輪郭抽出について説明します。輪郭は画像のエッジ部分を抽出することで得られます。エッジ部は

〈写真6〉
拡大表示例

〈２次元周波数特性〉

【図11】
水平垂直２倍の
拡大表示に使用する
２次元補間フィルタの特性

この２次元フィルタ特性は水平周波数と垂直周波数で変数分離できるため、
同じタップ係数の水平フィルタと垂直フィルタを縦続接続することで得られる。

タップ係数　［垂直フィルタ：３１タップ，水平フィルタ：３１タップ］

水平タップ番号	タップ係数	水平タップ番号	タップ係数
16	0.5003680626	8,24	0.0000000000
15,17	0.3158955788	7,25	0.0170634150
14,18	0.0000000000	6,26	0.0000000000
13,19	-0.0984441075	5,27	-0.0092475416
12,20	0.0000000000	4,28	0.0000000000
11,21	0.0514790416	3,29	0.0044533278
10,22	0.0000000000	2,30	0.0000000000
9,23	-0.0296845509	1,31	-0.0016991946

【図12】
テレビ画面を取り込み処理した後，再びテレビ画面に表示するための流れ

高域成分を含んでいるため，**図17**に示す2次元ハイパスフィルタを通し，絶対値に変換した後，2値化すれば得られます。この処理の様子を**図18**に示します。

また実際に輪郭を抽出した例として**写真14**に原画を，**写真15**に輪郭抽出後の画像を示します。この他にもフィルタ処理の応用として，ソフトフォーカス処理や，輪郭強調処理などがあります。

各種テスト信号について

コンピュータでデータを作成しフィールドメモリーに転送することで，各種のテスト信号を作成することができます。以下にいくつかの例を示します。

①ゾーンプレート信号

ゾーンプレート信号は2次元周波数特性を観察するために，広く用いられている信号です。この信号を使用すると，NTSCデコーダ(テレビ受像機)のY/C分離特性がよくわかります。ゾーンプレート信号は，(2)式で表現されます。

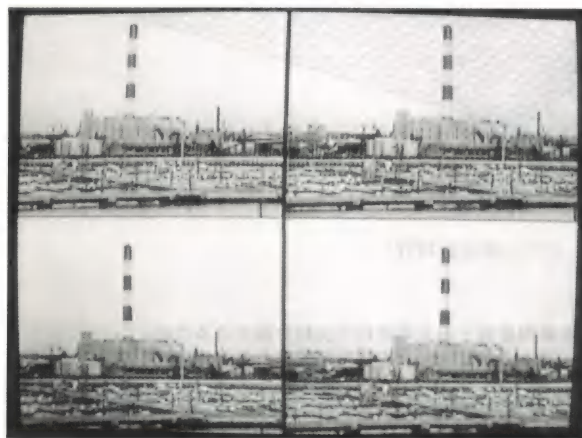
$$A(x, y) = A_0\cos\left(\frac{\pi}{648}x^2 + \frac{\pi}{480}y^2\right) + B \quad \cdots\cdots(2)$$

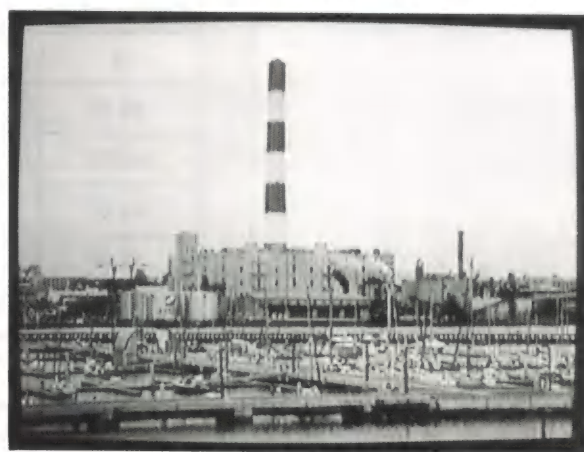
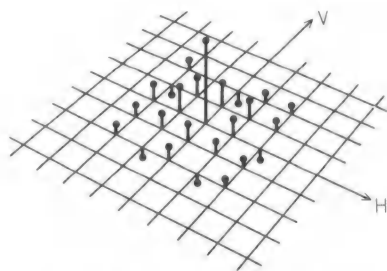
x＝－376～376：中心からの距離(ドット数)

y＝－240～240：中心からの距離(ライン数)

A_0：信号の振幅，B：中間レベル

②鋸歯状波信号(SAWTOOTH)

ガンマ特性や非直線性を確認したり，ディジタル映像機器のラッチミスなどの不具合を見つけるのに便利です。**写真16**に鋸歯状波信

〈写真7〉
縮小4画面表示例

〈写真8〉 2倍拡大表示例

〈写真9〉 写真7・8の原画

【図13】 2次元インパルス応答の例

号を示します。

③マルチバースト信号

伝送路のF特を確認するのに便利な信号です。**写真17**にマルチバースト信号の写真を示します。

その他にも，非常に多くのテスト信号や標準画像があります。基本的に，静止画であればこの多機能フレームメモリーによって発生することができます。

モノスコパターンや，一般画像はパソコンでデータを発生させることはできませんが，多機能フレ

〈写真10〉 ゾーンプレート

〈 2次元周波数特性〉

タップ係数　　　ＦＩＲフィルタ［垂直１タップ，水平３１タップ］

水平タップ番号	タップ係数	水平タップ番号	タップ係数
16	0.4484971016	8,24	-0.0215766796
15,17	0.3111540776	7,25	0.0022910309
14,18	0.0482025056	6,26	0.0126617534
13,19	-0.0871783396	5,27	0.0016887067
12,20	-0.0413377455	4,28	-0.0061474890
11,21	0.0358675822	3,29	-0.0021422958
10,22	0.0317604045	2,30	0.0022555776
9,23	-0.0129896478	1,31	0.0012420080

【図14】 2次元フィルタの特性-I　水平フィルタ

〈写真11〉 フィルタリング例①
(図14のフィルタ使用)

〈写真12〉 フィルタリング例②
(図15のフィルタ使用)

〈写真13〉 フィルタリング例③
(図16のフィルタ使用)

ームメモリーに入力し，パソコンに転送することでファイル化し，必要な時にいつでも呼び出すことが可能です。

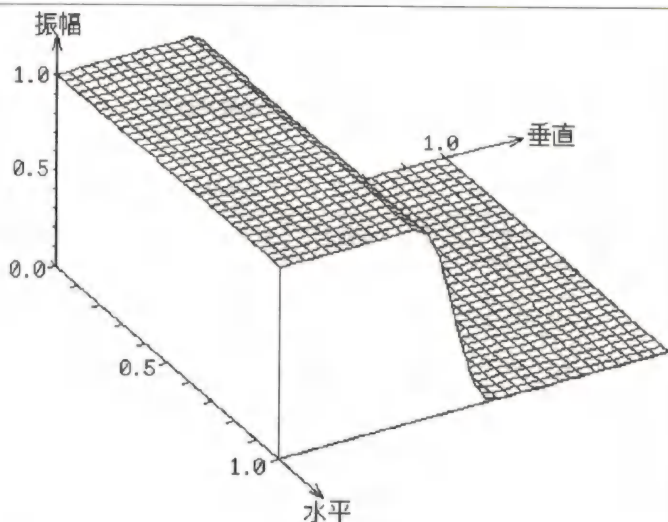
ハードウェアの変更点について

前回までで，ハードウェアの説明は終了しています。しかし，パソコンと多機能フレームメモリーの間で画像データの転送を行う場合に，何点か問題が生じたために，ハードウェアの変更を行いました。その内容は以下の通りです。

①フィールドインデックス信号発生回路の変更

今回製作した多機能フレームメモリーでは，フレームメモリー回路の制御に必要なRE・WE・RSTR・RSTW信号を発生するために，フィールドインデックス信号(EVEN/ODD)を使用しています。このEVEN/ODDは当初，同期分離用IC(LM1881)から出力されるO/E信号をLS244でバッファして，そのままEVEN/ODD信号として使用していました。

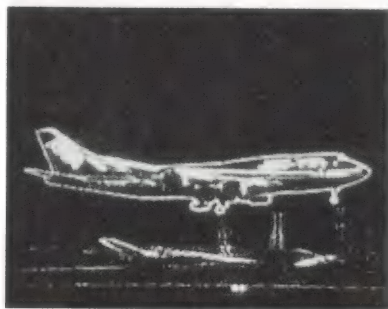
しかし，同期分離用IC(LM1881)がアナログICであるため，EVEN/ODD信号のデューティが完全に50%ではなく，そのため第一フィールドのサンプル数と，第二フィールドのサンプル数に若干の相違が生じます。試作した回路では，27サンプルの差が生じました。この差はソフトウェアで対処してもよいのですが，入力する映像信号の内容により多少変動するため，入力する映像信号が変わるたびに，パラメタを変更するわずらわしさがあるので，回路を変更しました。

図19に回路変更の内容を示します。主な変更点はソニー製の同期発生用IC(CX7930A)を追加したことです。**図20**にCX7930Aのブロック図を示します。また，実装した

〈 2次元周波数特性〉

タップ係数　　FIRフィルタ［水平1タップ，垂直31タップ］

垂直タップ番号	タップ係数	垂直タップ番号	タップ係数
16	0.4484971016	8,24	-0.0215766796
15,17	0.3111540776	7,25	0.0022910309
14,18	0.0482025056	6,26	0.0126617534
13,19	-0.0871783396	5,27	0.0016887067
12,20	-0.0413377455	4,28	-0.0061474890
11,21	0.0358675822	3,29	-0.0021422958
10,22	0.0317604045	2,30	0.0022555776
9,23	-0.0129896478	1,31	0.0012420080

【図15】 2次元フィルタの特性-II　垂直フィルタ

〈写真14〉 輪郭抽出例の原画

〈写真15〉 輪郭抽出例

CX7930Aの様子を**写真18**に示します。このICも秋葉原で入手することができました。

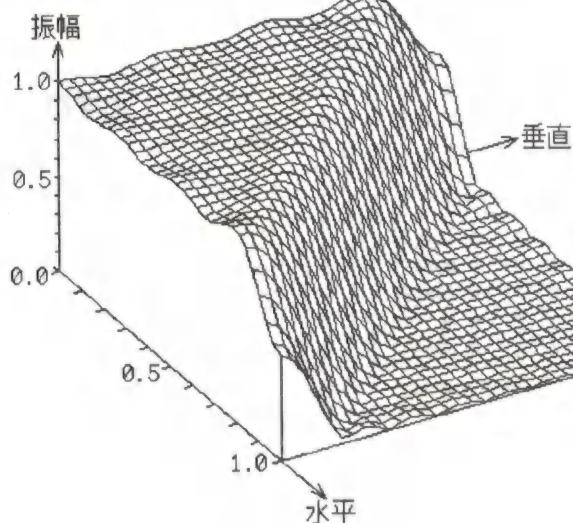
このICの5番ピン(OFLD)より出力される信号をEVEN/ODD信号として使用しました。このICを外部同期モードで使用し，クロック入力(CLIN)には14.3MHzの再生クロック信号を供給し，水平リセット入力(HRI)には，V7021より出力されるバーストフラグ(BF)信号をLS244でバッファして供給しました。また垂直リセット入力(VRI)には，LM1881より出力される垂直同期(VSYNC)信号を供給しました。

この回路変更によって第一フィールドと第二フィールドのサンプル数の相違はなくなりました。

②パソコンインタフェース回路の変更

最初，パソコンと信号の受け渡しをするインタフェース回路は，LS244のバッファ&ゲートICを使用し，3.3KΩのラダー抵抗でプルアップするだけの簡単な構成でした。しかし，実際にパソコンと接続しデータの転送を行うといくつかの問題が生じました。

まず，第一に電源ショックに弱いことです。静止画モード中でも，電源ショックでフレームメモリーの一部に書き込みが行われることが何度かありました。第二にパソコンからフレームメモリーにデータを転送する場合，8ビットのライトデータを変化させると，EXT-RD-CK信号やEXT-WR-CK信号，または，その他の制御信号ラインにスパイク状のノイズが発生し，誤動作します。

振幅
1.0
0.5
0.0
垂直
0.5
1.0
水平

〈2次元周波数特性〉

--------------タップ係数：水平15タップ、垂直15タップ-------------
(垂直、水平)　対称フィルタなので第1象現のみを示す。

係数	値	係数	値	係数	値
(0, 0)	0.493138 [センタタップ]				
(0, 1)	0.194364	(2, 6)	-0.000035	(5, 3)	0.000038
(0, 2)	-0.000056	(2, 7)	0.002149	(5, 4)	0.012528
(0, 3)	0.019523	(3, 0)	0.019523	(5, 5)	-0.001871
(0, 4)	-0.000048	(3, 1)	0.000052	(5, 6)	-0.007477
(0, 5)	0.005717	(3, 2)	0.033333	(5, 7)	0.000020
(0, 6)	-0.000037	(3, 3)	-0.002964	(6, 0)	-0.000037
(0, 7)	0.002112	(3, 4)	-0.020261	(6, 1)	-0.003475
(1, 0)	0.194364	(3, 5)	0.000038	(6, 2)	-0.000035
(1, 1)	-0.003643	(3, 6)	-0.003971	(6, 3)	-0.003971
(1, 2)	-0.061600	(3, 7)	0.000027	(6, 4)	-0.000029
(1, 3)	0.000052	(4, 0)	-0.000048	(6, 5)	-0.007477
(1, 4)	-0.010564	(4, 1)	-0.010564	(6, 6)	0.001295
(1, 5)	0.000043	(4, 2)	-0.000046	(6, 7)	0.004080
(1, 6)	-0.003475	(4, 3)	-0.020261	(7, 0)	0.002112
(1, 7)	0.000031	(4, 4)	0.002445	(7, 1)	0.000031
(2, 0)	-0.000056	(4, 5)	0.012528	(7, 2)	0.002149
(2, 1)	-0.061600	(4, 6)	-0.000029	(7, 3)	0.000027
(2, 2)	0.003377	(4, 7)	0.002370	(7, 4)	0.002370
(2, 3)	0.033333	(5, 0)	0.005717	(7, 5)	0.000020
(2, 4)	-0.000046	(5, 1)	0.000043	(7, 6)	0.004080
(2, 5)	0.006421	(5, 2)	0.006421	(7, 7)	-0.000765

【図16】2次元フィルタの特性-III　ダイアゴナルフィルタ

〈写真16〉鋸歯状波信号

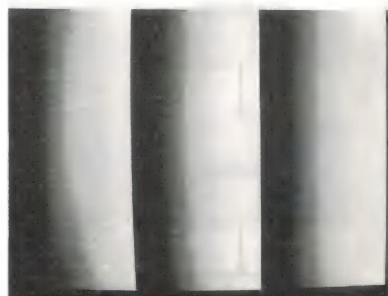
〈写真17〉マルチバースト信号

--------------タップ係数：水平７タップ、垂直７タップ------------

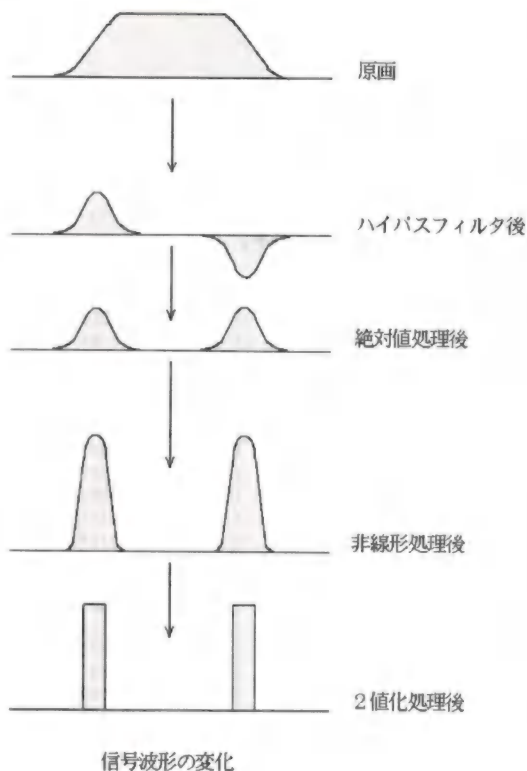
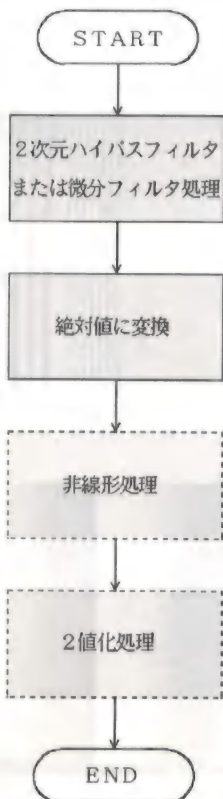
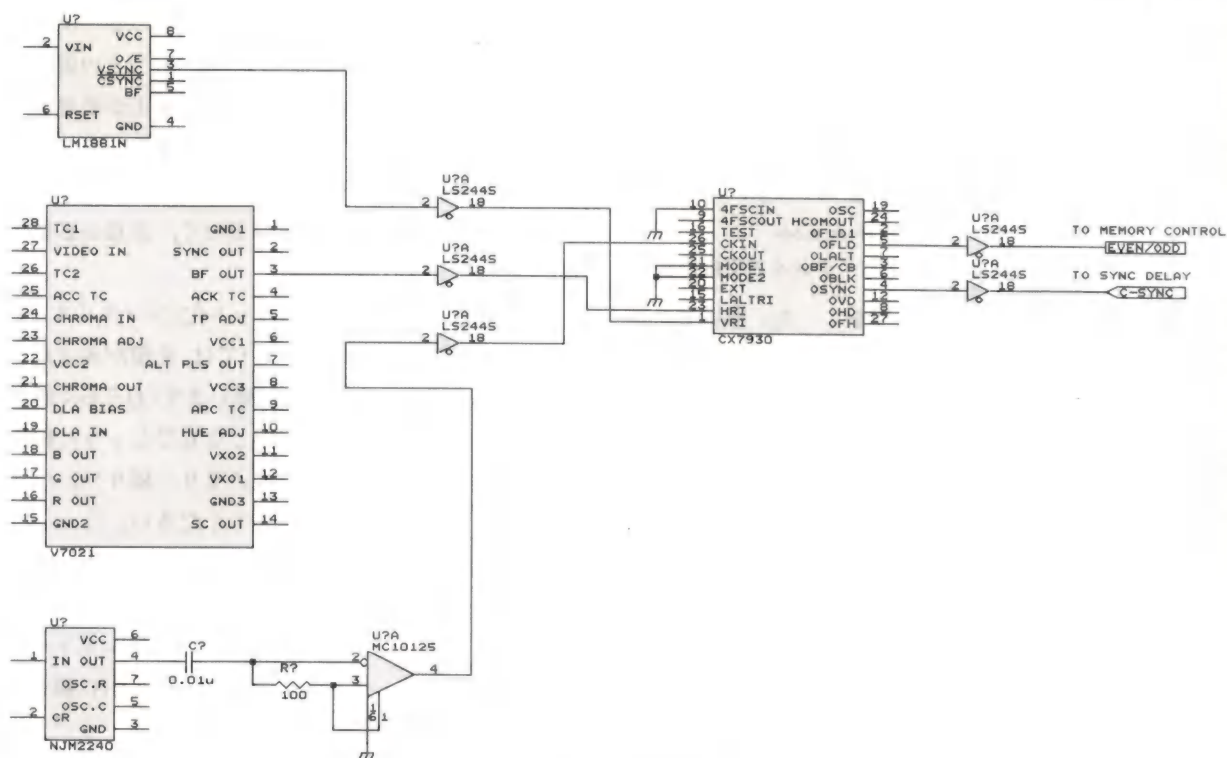
（垂直、水平） 対称フィルタなので第１象現のみを示す。

(0, 0) 0.910936[センタタップ]

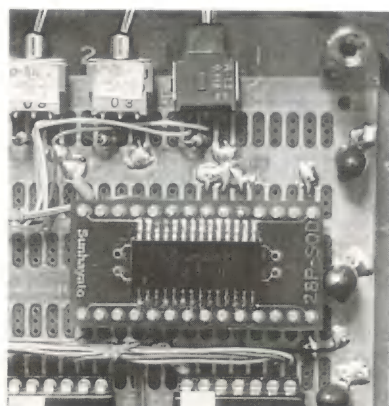
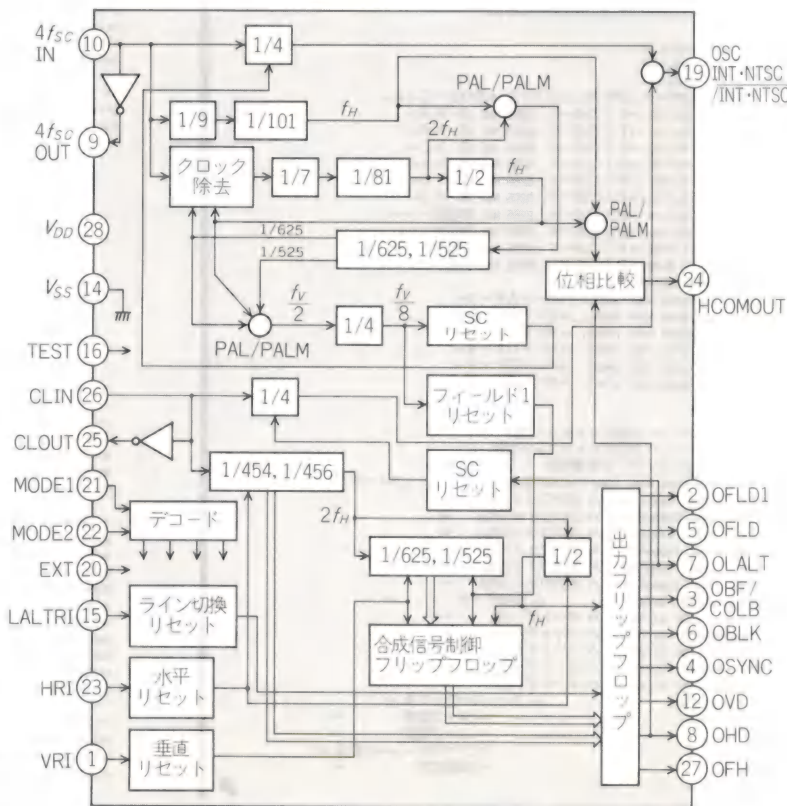
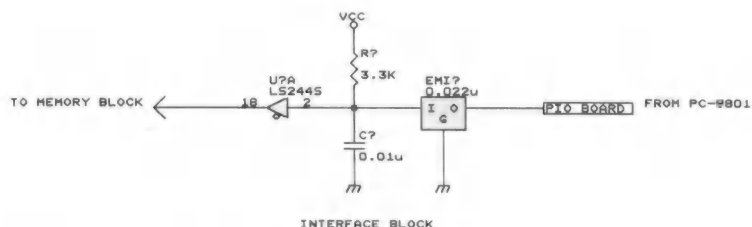
(0, 1) -0.069393	(1, 2) -0.023673	(2, 3) -0.001675
(0, 2) -0.030383	(1, 3) -0.003825	(3, 0) -0.004910
(0, 3) -0.004910	(2, 0) -0.030383	(3, 1) -0.003825
(1, 0) -0.069393	(2, 1) -0.023673	(3, 2) -0.001675
(1, 1) -0.054067	(2, 2) -0.010365	(3, 3) -0.000271

【図17】 輪郭抽出に使用される２次元ハイパスフィルタの特性

これらの問題を解決するため，**図21**に示すようにスパイク状のノイズを吸収する回路を追加しました。今回使用したフィールドメモリー用ICのTMS4C1050は非常に高速動作をするために，ちょっとしたヒゲにも反応してしまいます。追加した回路は一種のLPFですから，動作クロック周波数が非常に高いパソコンを使用し誤動作を生じた場合には，追加した回路の定数を変更してみてください。

むすび

今回で，多機能フレームメモリーの製作は一応終了です。ハードウェア・ソフトウェアともに十分に説明できなかった部分も多々ありましたが，なるべく読者のみな

【図18】
輪郭抽出の処理の流れ

【図19】 回路の変更部

さんが製作されるときの助けとなることを考慮して，執筆したつもりです。

この多機能フレームメモリーは，パソコンしか持たない一般のアマチュアの方が，画像処理を実際に自分の手で行い，どのような処理をすれば，画像がどのように変化するのかを体験してもらうために企画しました。

〈写真18〉 CX-7930A

【図20】 CX-7930Aのブロック図

【図21】スパイクノイズ吸収回路

パソコンですから計算時間は長くなりますが，ソフトウェアを作成すれば，ベクトル量子化・DCT(離散コサイン変換)・ウエーブレット変換・形状解析などいろいろな処理が可能です。筆者もこの原稿を書きながら，色々な画像処理の実験をしています。

今回の製作で一番苦労したのは，やはり部品の調達です。規格表やカタログに掲載されていても，入手できない部品がかなりありました。部品集めをしていたのがスキーシーズンで，この原稿を書いている今は，梅雨の季節ですから時間の経つのは早いものです。それでもなんとか完成にこぎつけて，筆者もホッとしています。

紙面の都合もあり，少ししかプログラムリストを載せることができませんでしたが，基本的なC言語のソースプログラム例をご覧になって，自分で考えてみてください。

ただし，使用しているパソコン環境(機種やメモリーなどの周辺部など)によっては，修正が必要となることも考えられます。そういったところはそれぞれのシステムによって異なるので，注意が必要です。

部品のキット化(ソースリストを含む)については，希望が多ければ考えたいと思います(おそらく価格は10数万円となるでしょう)。編集部フレームメモリー係りへ，ハガキで連絡してください。

なお，本製作のプログラムについてC言語のソースリストをパソコン通信PC-VANのエレクトロニクスライフSIG(JELEX)に載せることも考えています。

```
/*
****************************************
* 映像データ取得プログラム LOAD.C *
****************************************
*/
#include <stdio.h>
#include <conio.h>
#include <time.h>
/*使用ＰＩＯボード..アドテックシステム ＡＢ９８－０４Ａ*/
#define PA 0xffd8 /*ＰＩＯボード..8255 No2 ポートＡ*/
#define PB 0xffda /*ＰＩＯボード..8255 No2 ポートＢ*/
#define PC 0xffdc /*ＰＩＯボード..8255 No2 ポートＣ*/
#define PD 0xffd0 /*ＰＩＯボード..8255 No1 ポートＡ*/
#define PE 0xffd2 /*ＰＩＯボード..8255 No1 ポートＢ*/
#define PF 0xffd4 /*ＰＩＯボード..8255 No1 ポートＣ*/
#define C1 0xffd6 /*ＰＩＯボード..8255 No1 制御ポート*/
#define C2 0xffde /*ＰＩＯボード..8255 No2 制御ポート*/
#define CTL1 0x80 /*ＰＩＯボード..8255 No1 制御ワード*/
#define CTL2 0x90 /*ＰＩＯボード..8255 No2 制御ワード*/
#define FIELDLEN 241150
static unsigned char padat; /*ポートＡデータ*/
static unsigned char pbdat; /*ポートＢデータ*/
static unsigned char pcdat; /*ポートＣデータ*/
static unsigned char pddat; /*ポートＤデータ*/
static unsigned char pedat; /*ポートＥデータ*/
static unsigned char pfdat; /*ポートＦデータ*/
main()
{
void setbit(); /* 特定ビットセット関数    */
void rstbit(); /* 特定ビットリセット関数 */
void dataget();/* データ取得関数          */
static unsigned char huge field[FIELDLEN];     /* １フィールドデータ */
static char fname1[20]; /* 出力ファイル名1     */
static char fname2[20]; /* 出力ファイル名2     */
static char fname3[20]; /* 出力ファイル名3     */
static char fname4[20]; /* 出力ファイル名4     */
static char fname5[20]; /* 出力ファイル名5     */
static char fname6[20]; /* 出力ファイル名6     */
unsigned char data;              /* 映像データ          */
unsigned char buff0;    /* ポートデータ0     */
unsigned char buff1;    /* ポートデータ1     */
unsigned char bufe0;    /* ポートデータ2     */
unsigned char bufe1;    /* ポートデータ3     */
unsigned char bufe2;    /* ポートデータ4     */
long fieldlen;                   /* フィールドデータ長 */
long loop;                                /* ループ変数       */
int  i,j,k;                               /* ループ変数       */
int  ch;                                  /* データチャンネル番号 */
char cc;                                  /* 制御文字         */

FILE *fopen(),*fp1;

        /* スタートメッセージ */
        printf("¥n3CH_フレームデータ取得プログラム LOAD.C");
        printf("¥nＰＩＯボード基本アドレス=FFD0");
        fieldlen=FIELDLEN;
        printf("¥nFIELDLEN=%ld",FIELDLEN);

        /* ファイル名の指定 */
        printf("¥n出力データファイルのファイル名を指定して下さい。!!");
        do{
                printf("¥nＲ_フィールド１ファイル名=");
                scanf("%s",fname1);
                printf( "Ｒ_フィールド２ファイル名=");
                scanf("%s",fname2);
                printf( "Ｇ_フィールド１ファイル名=");
                scanf("%s",fname3);
                printf( "Ｇ_フィールド２ファイル名=");
                scanf("%s",fname4);
                printf( "Ｂ_フィールド１ファイル名=");
                scanf("%s",fname5);
                printf( "Ｂ_フィールド２ファイル名=");
                scanf("%s",fname6);
                printf("ＯＫですか？(y/n) ");
                cc=getche();
         }while((cc!='y')&&(cc!='Y'));

        /* ＰＩＯボード初期設定 */
        data=CTL1;
        outp(C1,CTL1);
        data=CTL2;
        outp(C2,CTL2);

        /* 全出力データを１にセット */
        pbdat=0xff;
        outp(PB,pbdat);
        pcdat=0xff;
        outp(PC,pcdat);
        pddat=0xff;
        outp(PD,pddat);
        pedat=0xff;
        outp(PE,pedat);
        pfdat=0xff;
        outp(PF,pfdat);

        /* 準備ＯＫの表示 */
        printf("¥n**データ取得の準備ができました**");
        printf("¥n本体のREMOT/LOCALスイッチをREMOTEにして何かキーを押してください。");
        cc=getch();
```

【リスト１】LOAD.C

```
        /* リモートゲートをONにする */
        rstbit(&pfdat,8);
        outp(PF,pfdat);
        /* リモートスチルをONにする。 */
        rstbit(&pddat,7);
        outp(PD,pddat);
        /* 画像静止トリガを待つ */
        do{
                setbit(&pddat,8);
                outp(PD,pddat);
                printf("¥n画像を静止させます。何かキーを押してください。");
                cc=getch();
                /* リモートエフェクトをONにする */
                rstbit(&pddat,8);
                outp(PD,pddat);
                printf("¥nOKですが？(y/n) ");
                cc=getche();
        }while((cc!='y')&&(cc!='Y'));

        for(ch=1;ch<=6;ch++){
                printf("¥nデータ取得中!!");
                dataget(field,fieldlen,ch);
                /* 映像データのファイル出力 */
                printf("¥nファイル出力中!!");
                switch(ch){
                        case 1 : fp1=fopen(fname1,"wb");
                        break;
                        case 2 : fp1=fopen(fname2,"wb");
                        break;
                        case 3 : fp1=fopen(fname3,"wb");
                        break;
                        case 4 : fp1=fopen(fname4,"wb");
                        break;
                        case 5 : fp1=fopen(fname5,"wb");
                        break;
                        case 6 : fp1=fopen(fname6,"wb");
                        break;
                        defualt:break;
                }
                for(loop=0;loop<fieldlen;loop++) putc(field[loop],fp1);
                fclose(fp1);
        }
        printf("¥nプログラム終了!!");
}

/*
****************************************
*       特定ビットセット関数            *
* 指定したビットだけ1にセットします *
****************************************
*/
static unsigned char bit[9]={0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
void setbit(byte,bitn)
unsigned char *byte;
int bitn;
{
unsigned char buff;

        buff=*byte;
        *byte=buff|bit[bitn];
}

/*
*******************************************
*       特定ビットリセット関数              *
* 指定したビットだけ0にリセットします。*
*******************************************
*/
void rstbit(byte,bitn)
unsigned char *byte;
int bitn;
{
unsigned char buff;

        buff=*byte;
        *byte=buff&(bit[bitn]^0xff);
}

/*
****************
* 時間待ち関数 *
****************
*/
void delay()
{
time_t time1,time2;      /* 時刻変数 */

        time(&time1);
        do{
                time(&time2);
        }while(time2-time1<1);
}

/*フィールドデータ取得関数*/
void dataget(field_dat,fieldlen,ch)
unsigned char huge field_dat[]; /*1フィールドデータ   */
                long fieldlen; /*フィールドデータ長  */
                int  ch;       /*取得データチャンネル*/
/*
    ch=1:Rch_field1_data
    ch=2:Rch_field2_data
    ch=3:Gch_field1_data
    ch=4:Gch_field2_data
    ch=5:Bch_field1_data
    ch=6:Bch_field2_data
*/
{
void setbit();  /* 特定ビットセット関数   */
void rstbit();  /* 特定ビットリセット関数 */
void delay();   /* 時間待ち               */
unsigned char data;              /* 映像データ    */
unsigned char buff0;     /* ポートデータ0 */
unsigned char buff1;     /* ポートデータ1 */
unsigned char bufe0;     /* ポートデータ2 */
unsigned char bufe1;     /* ポートデータ3 */
unsigned char bufe2;     /* ポートデータ4 */
                long loop;                  /* ループ変数     */

        /*初期状態の作成・設定*/
        /*リモートリードゲートを開く*/
        switch(ch){
                case 1 :rstbit(&pfdat,1);break;
                case 2 :rstbit(&pfdat,1);break;
                case 3 :rstbit(&pfdat,2);break;
                case 4 :rstbit(&pfdat,2);break;
                case 5 :rstbit(&pfdat,3);break;
                case 6 :rstbit(&pfdat,3);break;
                defualt:break;
        }
        outp(PF,pfdat);
        /*メモリリード・ライト制御信号を初期化する*/
        rstbit(&pedat,3);
        rstbit(&pedat,4);
        rstbit(&pedat,7);
        rstbit(&pedat,8);
        rstbit(&pedat,1);
        rstbit(&pedat,2);
        rstbit(&pedat,5);
        rstbit(&pedat,6);
        outp(PE,pedat);
        bufe0=pedat;
        /*外部クロック信号を初期化する*/
        rstbit(&pfdat,4);
        rstbit(&pfdat,5);
        outp(PF,pfdat);
        /*制御ワードの事前作成*/
        if((ch==1)||(ch==3)||(ch==5)){
                /*フィールド1リードイネーブル*/
                setbit(&pedat,3);
                /*リードポインタ1リセットアクティブ*/
                setbit(&pedat,7);
                /*ライトポインタ1リセットアクティブ*/
                setbit(&pedat,5);
                bufe1=pedat;
                /*リードポインタ1リセットディスアクティブ*/
                rstbit(&pedat,7);
                /*ライトポインタ1リセットディスアクティブ*/
                rstbit(&pedat,5);
                bufe2=pedat;
        }
        else{
                /*フィールド2リードイネーブル*/
                setbit(&pedat,4);
                /*リードポインタ2リセットアクティブ*/
                setbit(&pedat,8);
                /*ライトポインタ2リセットアクティブ*/
                setbit(&pedat,6);
                bufe1=pedat;
                /*リードポインタ2リセットをディスアクティブにする*/
                rstbit(&pedat,8);
                /*ライトポインタ2リセットをディスアクティブにする*/
                rstbit(&pedat,6);
                bufe2=pedat;
        }
        /*リードクロックHデータ*/
        setbit(&pfdat,4);
        setbit(&pfdat,5);
        buff1=pfdat;
        /*リードクロックLデータ*/
        rstbit(&pfdat,4);
        rstbit(&pfdat,5);
        buff0=pfdat;

        /************************** データの取得 **************************/
        switch(ch){
                case 1 :printf("¥nRchフィールド1データ取得開始!!");
                break;
                case 2 :printf("¥nRchフィールド2データ取得開始!!");
                break;
                case 3 :printf("¥nGchフィールド1データ取得開始!!");
                break;
                case 4 :printf("¥nGchフィールド2データ取得開始!!");
                break;
                case 5 :printf("¥nBchフィールド1データ取得開始!!");
                break;
                case 6 :printf("¥nBchフィールド2データ取得開始!!");
                break;
```

【リスト1】LOAD.Cつづき

```
		default:break;
	}
	/*メモリ制御を外部にする*/
	if((ch==1)||(ch==2)){
		rstbit(&pddat,1);
		rstbit(&pddat,2);
	}
	else if((ch==3)||(ch==4)){
		rstbit(&pddat,3);
		rstbit(&pddat,4);
	}
	else{
		rstbit(&pddat,5);
		rstbit(&pddat,6);
	}
	outp(PD,pddat);
	/*時間待ち*/
	delay();
	/*メモリクロックを外部にする*/
	if((ch==1)||(ch==2)) rstbit(&pcdat,4);
	else if((ch==3)||(ch==4)) rstbit(&pcdat,5);
	else rstbit(&pcdat,6);
	outp(PC,pcdat);
	loop=0;
	/*リードイネーブル&メモリポインタリセットをアクティブにする*/
	outp(PE,bufe1);
	/*リード・ライトクロックをHにする*/
	outp(PF,buff1);
	/*映像データを読み込む*/
	data=inp(PA);
	field_dat[loop]=data;
	/*リードライトクロックをLにする*/
	outp(PF,buff0);
	/*リード・ライトポインタリセットをディスアクティブにする*/
	outp(PE,bufe2);
	/*データ取得ループ*/
	for(loop=1;loop<fieldlen;loop++){
		/*リード・ライトクロックをHにする*/
		outp(PF,buff1);
		/*映像データを読み込む*/
		data=inp(PA);
		field_dat[loop]=data;
		/*リード・ライトクロックをLにする*/
		outp(PF,buff0);
	};

	/*================== 後処理 =================*/
	/*リードイネーブルをディスアクティブにする*/
	outp(PE,bufe0);
	/*リード・ライトクロックをHにする*/
	outp(PF,buff1);
	/*リードライトクロックをLにする*/
	outp(PF,buff0);
	/*メモリクロックを内部にする*/
	if((ch==1)||(ch==2)) setbit(&pcdat,4);
	else if((ch==3)||(ch==4)) setbit(&pcdat,5);
	else setbit(&pcdat,6);
	outp(PC,pcdat);
	/*メモリ制御を内部にする*/
	if((ch==1)||(ch==2)){
		setbit(&pddat,1);
		setbit(&pddat,2);
	}
	else if((ch==3)||(ch==4)){
		setbit(&pddat,3);
		setbit(&pddat,4);
	}
	else{
		setbit(&pddat,5);
		setbit(&pddat,6);
	}
	outp(PD,pddat);
	/*リード・ライトクロックをHにする*/
	outp(PF,buff1);
	/*リモートリードゲートを閉じる*/
	switch(ch){
		case 1 :setbit(&pfdat,1);break;
		case 2 :setbit(&pfdat,1);break;
		case 3 :setbit(&pfdat,2);break;
		case 4 :setbit(&pfdat,2);break;
		case 5 :setbit(&pfdat,3);break;
		case 6 :setbit(&pfdat,3);break;
		default:break;
	}
	outp(PF,pfdat);
}
```

【リスト1】LOAD.Cおわり

```
/*
*************************************************************
*              映像データ転送プログラム SEND.C               *
* 3チャンネル分のバイナリファイルデータをフレームメモリへ転送する *
*************************************************************
*/
#include <stdio.h>
#include <conio.h>
#include <time.h>
/* 使用PIOボード..アドテックシステム AB98-04A */
#define PA 0xffd8     /* PIOボード..8255 No2 ポートA */
#define PB 0xffda         /* PIOボード..8255 No2 ポートB */
#define PC 0xffdc         /* PIOボード..8255 No2 ポートC */
#define PD 0xffd0         /* PIOボード..8255 No1 ポートA */
#define PE 0xffd2         /* PIOボード..8255 No1 ポートB */
#define PF 0xffd4         /* PIOボード..8255 No1 ポートC */
#define C1 0xffd6         /* PIOボード..8255 No1 制御ポート */
#define C2 0xffde         /* PIOボード..8255 No2 制御ポート */
#define CTL1 0x80         /* PIOボード..8255 No1 制御ワード */
#define CTL2 0x90         /* PIOボード..8255 No2 制御ワード.*/
#define FIELDLEN 241150   /* 1フィールド総サンプル数   */
static unsigned char padat;             /* ポートAデータ */
static unsigned char pbdat;             /* ポートBデータ */
static unsigned char pcdat;             /* ポートCデータ */
static unsigned char pddat;             /* ポートDデータ */
static unsigned char pedat;             /* ポートEデータ */
static unsigned char pfdat;             /* ポートFデータ */
main()
{
void    setbit();       /* 特定ビットセット関数   */
void    rstbit();       /* 特定ビットリセット関数 */
void    datasnd();      /* データ転送関数         */
static unsigned char huge field[FIELDLEN];    /* 1フィールドデータ */
static unsigned char data;      /* 映像データ     */
static unsigned char buff0;     /* ポートデータ0  */
static unsigned char buff1;     /* ポートデータ1  */
static unsigned char bufe0;     /* ポートデータ2  */
static unsigned char bufe1;     /* ポートデータ3  */
static unsigned char bufe2;     /* ポートデータ4  */
static int  ch;                         /* データチャンネル番号 */
static long fieldlen;           /* フィールドデータ長 */
static int  i,j,k;                      /* ループ変数     */
static long loop;                       /* ループ変数     */
static long pointer;            /* データポインタ  */
static char fname1[20];         /* 入力ファイル名1 */
static char fname2[20];         /* 入力ファイル名2 */
static char fname3[20];         /* 入力ファイル名3 */
static char fname4[20];         /* 入力ファイル名4 */
static char fname5[20];         /* 入力ファイル名5 */
static char fname6[20];         /* 入力ファイル名6 */
static char cc;                         /* 制御文字       */
FILE *fopen(),*fp1;                     /* ファイルポインタ */

	/* スタートメッセージ */
	printf("¥n3CHフレームデータ転送プログラム SEND.C");
	printf("¥nPIOボード基本アドレス=FFD0");
	fieldlen=FIELDLEN;
	printf("¥nFIELDLEN=%ld",FIELDLEN);

	/* ファイル名の指定 */
	printf("¥n転送データファイルのファイル名を指定して下さい。");
	do{
		printf("¥nR_フィールド1データファイル名=");
		scanf("%s",fname1);
		printf( "R_フィールド2データファイル名=");
		scanf("%s",fname2);
		printf( "G_フィールド1データファイル名=");
		scanf("%s",fname3);
		printf( "G_フィールド2データファイル名=");
		scanf("%s",fname4);
		printf( "B_フィールド1データファイル名=");
		scanf("%s",fname5);
		printf( "B_フィールド2データファイル名=");
		scanf("%s",fname6);
		printf("OKですか?(y/n) ");
		cc=getche();
	}while((cc!='y')&&(cc!='Y'));

	/* PIOボード初期設定 */
	data=CTL1;
	outp(C1,CTL1);
	data=CTL2;
	outp(C2,CTL2);

	/* 全出力データを1にセット */
	pbdat=0xff;
	outp(PB,pbdat);
	pcdat=0xff;
	outp(PC,pcdat);
	pddat=0xff;
	outp(PD,pddat);
	pedat=0xff;
	outp(PE,pedat);
	pfdat=0xff;
```

【リスト2】SEND.C

```c
        outp(PF,pfdat);

        /* 準備OKの表示 */
        printf("¥n***データ転送の準備ができました。***");
        printf("¥n本体のREMOT/LOCALスイッチをREMOTEにして何かキーを押してください。");
        cc=getch();

        /* リモートゲートをONにする */
        printf("¥nリモートゲートON!!");
        rstbit(&pfdat,8);
        outp(PF,pfdat);

        /* リモートスチルをONにする。 */
        printf("¥nリモ一トスチルON!!");
        rstbit(&pddat,7);
        outp(PD,pddat);

        /* リモートエフェクトをONにする */
        printf("¥nリモートエッフェクトON!!");
        rstbit(&pddat,8);
        outp(PD,pddat);

        for(ch=1;ch<=6;ch++){
                printf("¥nファイル読み込み中!!");
                switch(ch){
                        case 1 : fp1=fopen(fname1,"rb");
                        break;
                        case 2 : fp1=fopen(fname2,"rb");
                        break;
                        case 3 : fp1=fopen(fname3,"rb");
                        break;
                        case 4 : fp1=fopen(fname4,"rb");
                        break;
                        case 5 : fp1=fopen(fname5,"rb");
                        break;
                        case 6 : fp1=fopen(fname6,"rb");
                        break;
                        defualt:break;
                }
                /* データのファイル読み込み */
                for(pointer=0;pointer<fieldlen;pointer++) field[pointer]=getc(fp1);
                fclose(fp1);
                /* データの転送 */
                printf("¥nデータ転送中!!");
                datasnd(field,fieldlen,ch);
        }
        printf("¥n転送を終了しました。!!");
        printf("¥nプログラム終了!1");
}

/* 特定ビットセット関数 */
/* 指定したビットだけ1にセットします。 */
static unsigned char bit[9]={0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
void setbit(byte,bitn)
 unsigned char *byte;
 int bitn;
 {
 unsigned char buff;

        buff=*byte;
        *byte=buff|bit[bitn];
 }

/* 特定ビットリセット関数 */
/* 指定したビットだけ0にリセットします。 */
void rstbit(byte,bitn)
unsigned char *byte;
int bitn;
{
unsigned char buff;

        buff=*byte;
        *byte=buff&(bit[bitn]^0xff);
}

/* 時間待ち関数 */
void delay()
{
time_t time1,time2;     /* 時刻変数 */

        time(&time1);
        do{
                time(&time2);
        }while(time2-time1<1);
}

/* フィールドデータ転送関数 */
void datasnd(field_dat,fieldlen,ch)
unsigned char huge field_dat[];         /* 1フィールドデータ   */
long fieldlen;                          /* フィールドデータ長   */
int     ch;                             /* 取得データチャンネル */
/*
   ch=1:Rch_field1_data
   ch=2:Rch_field2_data
   ch=3:Gch_field1_data
   ch=4:Gch_field2_data
   ch=5:Bch_field1_data
   ch=6:Bch_field2_data
*/
{
```

```c
void setbit();  /* 特定ビットセット関数   */
void rstbit();  /* 特定ビットリセット関数 */
void delay();   /* 時間待ち               */
static unsigned char data;      /* 映像データ   */
static unsigned char buff0;     /* ポートデータ0 */
static unsigned char buff1;     /* ポートデータ1 */
static unsigned char bufe0;     /* ポートデータ2 */
static unsigned char bufe1;     /* ポートデータ3 */
static unsigned char bufe2;     /* ポートデータ4 */
static long loop;                       /* ループ変数   */

        /*   初期状態の作成・設定   */

        /* 内部ライトゲートを閉じる */
        rstbit(&pcdat,7);
        outp(PC,pcdat);

        /* 外部ライトゲートを開く */
        if((ch==1)||(ch==2)) rstbit(&pcdat,1);
        else if((ch==3)||(ch==4)) rstbit(&pcdat,2);
        else rstbit(&pcdat,3);
        outp(PC,pcdat);

        /* メモリリード・ライト制御信号を初期化する */
        /* 各チャンネル各フィールド共通 */
        rstbit(&pedat,3);
        rstbit(&pedat,4);
        rstbit(&pedat,7);
        rstbit(&pedat,8);
        rstbit(&pedat,1);
        rstbit(&pedat,2);
        rstbit(&pedat,5);
        rstbit(&pedat,6);
        outp(PE,pedat);
        bufe0=pedat;

        /* 外部クロック信号を初期化する */
        /* 各チャンネル各フィールド共通 */
        rstbit(&pfdat,4);
        rstbit(&pfdat,5);
        outp(PF,pfdat);

        /* 制御ワードの事前作成 */

        if((ch==1)||(ch==3)||(ch==5)){
                /* フィールド1ライトイネーブル */
                setbit(&pedat,1);
                /* リードポインタ1リセットアクティブ */
                setbit(&pedat,7);
                /* ライトポインタ1リセットアクティブ */
                setbit(&pedat,5);
                bufe1=pedat;
                /* リードポインタ1リセットをディスアクティブにする */
                rstbit(&pedat,7);
                /* ライトポインタ1リセットをディスアクティブにする */
                rstbit(&pedat,5);
                bufe2=pedat;
        }
        else{
                /* フィールド2ライトイネーブル */
                setbit(&pedat,2);
                /* リードポインタ2リセットアクティブ */
                setbit(&pedat,8);
                /* ライトポインタ2リセットアクティブ */
                setbit(&pedat,6);
                bufe1=pedat;
                /* リードポインタ2リセットをディスアクティブにする */
                rstbit(&pedat,8);
                /* ライトポインタ2リセットをディスアクティブにする */
                rstbit(&pedat,6);
                bufe2=pedat;
        }

        /* リード・ライトクロックHデータ */
        setbit(&pfdat,4);
        setbit(&pfdat,5);
        buff1=pfdat;

        /* リード・ライトクロックLデータ */
        rstbit(&pfdat,4);
        rstbit(&pfdat,5);
        buff0=pfdat;

        /* メモリ制御を外部にする */
        if((ch==1)||(ch==2)){
                rstbit(&pddat,1);
                rstbit(&pddat,2);
        }
        else if((ch==3)||(ch==4)){
                rstbit(&pddat,3);
                rstbit(&pddat,4);
        }
        else{
                rstbit(&pddat,5);
                rstbit(&pddat,6);
        }
        outp(PD,pddat);

        /* 時間待ち */
```

【リスト2】SEND.Cつづき

```
	delay();

	/* メモリクロックを外部にする */
	if((ch==1)||(ch==2)) rstbit(&podat,4);
	else if((ch==3)||(ch==4)) rstbit(&podat,5);
	else rstbit(&podat,6);
	outp(PC,podat);

	/* ループカウンタリセット */
	loop=0;

	/* ライトイネーブル&メモリポインタリセットをアクティブにする */
	outp(PE,bufe1);

	/* 映像データを出力する */
	data=field_dat[loop];
	outp(PB,data);

	/* リード・ライトクロックをHにする */
	outp(PF,buff1);

	/* リードライトクロックをLにする */
	outp(PF,buff0);

	/* リード・ライトポインタリセットをディスアクティブにする */
	outp(PE,bufe2);

	/* データ送出ループ */
	for(loop=1;loop<fieldlen;loop++){
		/* 映像データを出力する */
		data=field_dat[loop];
		outp(PB,data);
		/* リード・ライトクロックをHにする */
		outp(PF,buff1);
		/* リード・ライトクロックをLにする */
		outp(PF,buff0);
	}

	/* ライトイネーブルをディスアクティブにする */
	/* リード・ライトリセットをアクティブにする */
	pedat=bufe1;
	if((ch==1)||(ch==3)||(ch==5)){
		rstbit(&pedat,1);
		outp(PE,pedat);
	}
	else{
		rstbit(&pedat,2);
		outp(PE,pedat);
	}
	/* リード・ライトクロックをHにする */
	outp(PF,buff1);
	/* リード・ライトクロックをLにする */
	outp(PF,buff0);
	/* リード・ライトリセットをアクティブにする */
	outp(PE,bufe0);
	/* リード・ライトクロックをHにする */
	outp(PF,buff1);
	/* リード・ライトクロックをLにする */
	outp(PF,buff0);

	/* メモリクロックを内部にする */
	if((ch==1)||(ch==2)) setbit(&podat,4);
	else if((ch==3)||(ch==4)) setbit(&podat,5);
	else setbit(&podat,6);
	outp(PC,podat);

	/* メモリ制御を内部にする */
	if((ch==1)||(ch==2)){
		setbit(&pddat,1);
		setbit(&pddat,2);
	}
	else if((ch==3)||(ch==4)){
		setbit(&pddat,3);
		setbit(&pddat,4);
	}
	else{
		setbit(&pddat,5);
		setbit(&pddat,6);
	}
	outp(PD,pddat);

	/* リード・ライトクロックをHにする */
	outp(PF,buff1);

	/* リモートライトゲートを閉じる */
	switch(ch){
		case 1 :setbit(&podat,1);break;
		case 2 :setbit(&podat,1);break;
		case 3 :setbit(&podat,2);break;
		case 4 :setbit(&podat,2);break;
		case 5 :setbit(&podat,3);break;
		case 6 :setbit(&podat,3);break;
		default:break;
	}
	outp(PC,podat);

	/* 内部ライトゲートを開く */
	setbit(&podat,7);
	outp(PC,podat);
}
```

【リスト2】SEND.C おわり

```
/*
*************************************************************************
*                    標準画面表示プログラム.....CRTDSP.C                  *
* フィールドメモリファイルよりデータを読み込みCRTディスプレイに表示する。*
* 白黒16階調モード、標準16色モード、特別16色モードのどれかを選択       *
* 特別16色モード時はパレットデータファイルを指定                       *
* 入力データファイルは6個を指定する。                                  *
*************************************************************************
*/
#include <stdio.h>
#include <graph.h>
#include <math.h>
#define VLEN 200
#define HLEN 640
#define HSAMPLE 910
#define HOFFSET 200
#define VOFFSET 22
#define ODDID  3.5              /* 奇数フィールド開始点 ライン単位 */
#define EVENID 3.0              /* 偶数フィールド開始点 ライン単位 */
#define BLOCK 4                 /* 分割ブロック数                  */
#define BLKLN 50                /* 1ブロック中のライン数           */
#define BLKSIZE 45500   /* 910*50 1ブロックのサンプル数  */
#define DATALEN 50000   /* バッファメモリのサイズ */
main()
{
static unsigned char huge data1[DATALEN];    /* Rフィールド1データ */
static unsigned char huge data2[DATALEN];    /* Rフィールド2データ */
static unsigned char huge data3[DATALEN];    /* Gフィールド1データ */
static unsigned char huge data4[DATALEN];    /* Gフィールド2データ */
static unsigned char huge data5[DATALEN];    /* Bフィールド1データ */
static unsigned char huge data6[DATALEN];    /* Bフィールド2データ */
static unsigned char table[4096];            /* パレット番号変換テーブル */
static unsigned char rr[16];    /* パレットデータR */
static unsigned char gg[16];    /* パレットデータG */
static unsigned char bb[16];    /* パレットデータB */
static char fname1[20]; /* 入力ファイル1の名前  */
static char fname2[20]; /* 入力ファイル2の名前  */
static char fname3[20]; /* 入力ファイル3の名前  */
static char fname4[20]; /* 入力ファイル4の名前  */
static char fname5[20]; /* 入力ファイル5の名前  */
static char fname6[20]; /* 入力ファイル6の名前  */
static char fnamep[20]; /* パレットデータファイルの名前 */
long far loop1; /* ループ変数       */
long far loop2; /* ループ変数       */
int i,j,k,l;    /* ループ変数       */
int mode;               /* 表示モード        */
int pltno1;             /* パレット番号1     */
int pltno2;             /* パレット番号2     */
int baseh;              /* 切りとり水平始点  */
int basev;              /* 切りとり垂直始点  */
int delt;               /*フィールド間の位相差*/
int ped;                /* 表示ペデスタル    */
long size;              /* データサイズ      */
double datr1;   /* 映像レベルR1     */
double datr2;   /* 映像レベルR2     */
double datg1;   /* 映像レベルG1     */
double datg2;   /* 映像レベルG2     */
double datb1;   /* 映像レベルB1     */
double datb2;   /* 映像レベルB2     */
double gain;    /* 表示ゲイン       */
double gainr;   /* 表示ゲインR      */
double gaing;   /* 表示ゲインG      */
double gainb;   /* 表示ゲインB      */
char  cc;               /* 制御文字          */
short wb16levpalette(); /* モノクロ16階調パレット設定関数 */
short standardpalette();/* カラー16色パレット設定関数     */
void readdat();                 /* データ読み込み関数   */
void mktable();                 /* 変換テーブル作成関数 */
void getpalt();                 /* パレット番号作成関数 */
FILE *fopen(),*fpp;
FILE *fp1,*fp2,*fp3;
FILE *fp4,*fp5,*fp6;

	_setvideomode(_98RESS16COLOR);
	printf("¥n**映像データ標準表示プログラム*CRTDSP**");
	/* 表示ページとアクティブページを設定する */
	_setactivepage(0);
	_setvisualpage(0);
```

【リスト3】CRTDSP.C

```
        /* 表示モードの決定 */
        printf("¥n表示モードを決定します。¥n");
        do{
                printf( "0=白黒16階調 1=16色標準 2=最適16色");
                printf("¥n表示モード= ");
                scanf("%d",&mode);
        }while((mode!=0)&&(mode!=1));
        /* カラーパレットを設定する */
        if(mode==1) standardpalette();
        else if(mode==2){
                printf("パレット設定用データファイル名を指定してください。");
                printf("¥nファイル名= ");
                scanf("%s",fnamep);
                fpp=fopen(fnamep,"r");
                for(i=0;i<16;i++) fscanf(fpp,"%d,%d,%d",rr[i],gg[i],bb[i]);
                fclose(fpp);
                mktable2(table,rr,gg,bb);
        }

        printf("映像データファイルを指定してください。");
        do{
                printf("¥nRchフィールド1データファイル名= ");
                scanf("%s",fname1);
                printf(  "Rchフィールド2データファイル名= ");
                scanf("%s",fname2);
                printf(  "Gchフィールド1データファイル名= ");
                scanf("%s",fname3);
                printf(  "Gchフィールド2データファイル名= ");
                scanf("%s",fname4);
                printf(  "Bchフィールド1データファイル名= ");
                scanf("%s",fname5);
                printf(  "Bchフィールド2データファイル名= ");
                scanf("%s",fname6);
                printf("よろしいですか？(y/n) ");
                cc=getch();
                putch(cc);
        }while((cc!='y')&&(cc!='Y'));

        do{
                fp1=fopen(fname1,"rb");
                fp2=fopen(fname2,"rb");
                fp3=fopen(fname3,"rb");
                fp4=fopen(fname4,"rb");
                fp5=fopen(fname5,"rb");
                fp6=fopen(fname6,"rb");

                /* 表示領域の設定 */
                printf("¥n切りとるテレビ画面の水平始点？= ");
                scanf("%d",&baseh);
                printf(  "切りとるテレビ画面の垂直始点？= ");
                scanf("%d",&basev);

                /* 表示オフセット分まで読み込む */
                printf("フィールド間始点誤差を指定してください。");
                printf("¥n誤差サンプル数= ");
                scanf("%d",&delt);
                printf("表示ゲインを指定してください。");
                if(mode==0){
                        printf("¥nゲイン= ");
                        scanf("%lf",&gain);
                }
                else{
                        printf("¥nRゲイン= ");
                        scanf("%lf",&gainr);
                        printf(  "Gゲイン= ");
                        scanf("%lf",&gaing);
                        printf(  "Bゲイン= ");
                        scanf("%lf",&gainb);
                }
                printf("表示ペデスタルを指定してください。");
                printf("¥nペデスタル= ");
                scanf("%d",&ped);

                size=(VOFFSET-ODDID+basev)*HSAMPLE+HOFFSET+baseh;
                readdat(fp1,data1,size);
                readdat(fp3,data3,size);
                readdat(fp5,data5,size);
                size=(VOFFSET-EVENID+basev)*HSAMPLE+HOFFSET+baseh+delt;
                readdat(fp2,data2,size);
                readdat(fp4,data4,size);
                readdat(fp6,data6,size);

                printf("表示を開始します。何かキーを押してください。!!");
                cc=getch();
                _clearscreen(_GCLEARSCREEN);
                if(mode==0){
                /* 白黒16階調 */
                        gain=gain/16.0;
                        for(k=0;k<BLOCK;k++){
                                /* バッファへ1ブロック分を読み込む */
                                readdat(fp1,data1,BLKSIZE);
                                readdat(fp2,data2,BLKSIZE);
                                readdat(fp3,data3,BLKSIZE);
                                readdat(fp4,data4,BLKSIZE);
                                readdat(fp5,data5,BLKSIZE);
                                readdat(fp6,data6,BLKSIZE);
                                l=2*k*BLKLN;
                                for(j=0;j<BLKLN;j++){
                                        for(i=0;i<HLEN;i++){
                                                loop1=(long)j*HSAMPLE+(long)i;
                                                loop2=(long)j*HSAMPLE+(long)i;
                                                datr1=data1[loop1]+ped;
                                                datr2=data2[loop2]+ped;
                                                datg1=data3[loop1]+ped;
                                                datg2=data4[loop2]+ped;
                                                datb1=data5[loop1]+ped;
                                                datb2=data6[loop2]+ped;
                                                pltno1=floor(gain*(0.3*datr1
                                                                +0.59*datg1
                                                                  +0.11*datb1)+0.5);
                                                pltno2=floor(gain*(0.3*datr2
                                                                +0.59*datg2
                                                                  +0.11*datb2)+0.5);
                                                if(pltno1>15) pltno1=15;
                                                if(pltno2>15) pltno2=15;
                                                _setcolor(pltno1);
                                                _setpixel(i+1,2*j+1+1);
                                                _setcolor(pltno2);
                                                _setpixel(i+1,2*j+2+1);
                                        }
                                }
                        }
                }
                else{
                        if(mode==1){
                                /* 標準16色 */
                                standardpalette();
                        else if(mode==2){
                                /* 特別16色 */
                                specialpalette(rr,gg,bb);
                        }
                        for(k=0;k<BLOCK;k++){
                                /* バッファへ1ブロック分を読み込む */
                                readdat(fp1,data1,BLKSIZE);
                                readdat(fp2,data2,BLKSIZE);
                                readdat(fp3,data3,BLKSIZE);
                                readdat(fp4,data4,BLKSIZE);
                                readdat(fp5,data5,BLKSIZE);
                                readdat(fp6,data6,BLKSIZE);
                                l=2*k*BLKLN;
                                for(j=0;j<BLKLN;j++){
                                        for(i=0;i<HLEN;i++){
                                                loop1=(long)j*HSAMPLE+(long)i;
                                                loop2=(long)j*HSAMPLE+(long)i;
                                                datr1=gainr*(double)data1[loop1]+(double)ped;
                                                datr2=gainr*(double)data2[loop2]+(double)ped;
                                                datg1=gaing*(double)data3[loop1]+(double)ped;
                                                datg2=gaing*(double)data4[loop2]+(double)ped;
                                                datb1=gainb*(double)data5[loop1]+(double)ped;
                                                datb2=gainb*(double)data6[loop2]+(double)ped;
                                                getpalt(datr1,datg1,datb1,table,&pltno1);
                                                getpalt(datr2,datg2,datb2,table,&pltno2);
                                                _setcolor(pltno1);
                                                _setpixel(i+1,2*j+1+1);
                                                _setcolor(pltno2);
                                                _setpixel(i+1,2*j+2+1);
                                        }
                                }
                        }
                }
                fclose(fp1);
                fclose(fp2);
                fclose(fp3);
                fclose(fp4);
                fclose(fp5);
                fclose(fp6);
                _settextposition(24,0);
                printf("描画を終了しました。OKですか？(y/n) ");
                cc=getch();
                _clearscreen(_GCLEARSCREEN);
        }while((cc!='y')&&(cc!='Y'));
}

/*
****************************************
* 標準16色モード時のパレット設定関数 *
****************************************
*/
short standardpalette()
{
static long far colors4096[]={   /* 4096色モード用配列  */
0x000000L/* 黒 */,0x0f0000L/* 青   */,0x00000fL/* 赤 */,0x0f000fL/* 紫 */,
0x000f00L/* 緑 */,0x0f0f00L/* 水色 */,0x000f0fL/* 黄 */,0x0f0f0fL/* 白 */,
0x010101L/*暗灰*/,0x080000L/* 暗青 */,0x000008L/*暗赤*/,0x080008L/*暗紫*/,
0x000800L/*暗緑*/,0x080800L/*暗水色*/,0x000808L/*暗黄*/,0x080808L/*暗白*/};
struct videoconfig far * config;

    _getvideoconfig(config);
    if((config->mode!=_98RESSCOLOR)
            &&(config->mode!=_98RESCOLOR)
                    &&(config->mode!=_98TEXT80))
                            return(_remapallpalette(colors4096));
}

/*
****************************************
* 白黒16階調モード時のパレット設定関数 *
****************************************
*/
```

【リスト3】CRTDSP.Cつづき

```
short wb16levpalette()
{
static long far mono16[]={/*モノクロ16階調用配列*/
0x000000L,0x010101L,0x020202L,0x030303L,
0x040404L,0x050505L,0x060606L,0x070707L,
0x080808L,0x090909L,0x0a0a0aL,0x0b0b0bL,
0x0c0c0cL,0x0d0d0dL,0x0e0e0eL,0x0f0f0fL};
return(_remapallpalette(mono16));
}

/*
****************************************
* 特別16色モード時のパレット設定関数 *
****************************************
*/
short specialpalette(rr,gg,bb)
unsigned char rr[];         /* パレットデータ赤 */
unsigned char gg[];     /* パレットデータ緑 */
unsigned char bb[];     /* パレットデータ青 */
{
static long far colors4096[16]; /* 4096色モード用配列 */
static unsigned long color1;    /* バッファ1         */
static unsigned long color2;    /* バッファ2         */
static unsigned long color3;    /* バッファ2         */
int     i,j,k;                                     /* ループパラメタ    */

        for(i=0;i<16;i++){
                color1=floor((double)rr[i]/16.0+0.5);
                color2=floor((double)gg[i]/16.0+0.5);
                color3=floor((double)bb[i]/16.0+0.5);
                color1=(0x00000fL)&color1;
                color2=(0x000f00L)&(color2<<8);
                color3=(0x0f0000L)&(color3<<16);
                colors4096[i]=color1|color2|color3;
        }
        return(_remapallpalette(colors4096));
}

/*
**********************
* データ読み込み関数 *
**********************
*/
void readdat(fp,data,size)
unsigned char huge data[]; /* データ */
FILE *fp;       /* データファイルポインタ */
long size;      /* 読み込みサイズ   */
{
long pointer; /* データポインタ */

        for(pointer=0;pointer<size;pointer++) data[pointer]=getc(fp);
}

/*
****************************************
* 映像データをパレット番号に変換する関数 *
****************************************
*/
#define REJECT 64.0 /* 6ビット分のリジェクト */
void getpalt(rdat,gdat,bdat,table,no)
double rdat;    /* Rチャンネルデータ */
double gdat;    /* Gチャンネルデータ */
double bdat;    /* Bチャンネルデータ */
unsigned char table[];/* 変換テーブル */
int *no;                    /* パレット番号       */
{
int     nbufr;          /* バッファ */
int     nbufg;          /* バッファ */
int     nbufb;          /* バッファ */

        nbuf1=floor(rdat/REJECT+0.5);
        if(nbuf1>3) nbuf1=3;
        nbuf2=floor(gdat/REJECT+0.5);
        if(nbuf2>3) nbuf2=3;
        nbuf3=floor(bdat/REJECT+0.5);
        if(nbuf3>3) nbuf3=3;
        *no=table[nbufr+4*nbufg+16*nbufb];
}
/*
**********************************
* パレット番号変換テーブル作成関数Ⅰ *
**********************************
*/
void mktable(table)
unsigned char table[]; /* パレット番号変換テーブル */
{
static unsigned char refr[16]; /* 基準信号R */
static unsigned char refg[16]; /* 基準信号G */
static unsigned char refb[16]; /* 基準信号B */
int r,g,b;                  /* ループパラメタ */
int i,j,k;                  /* ループパラメタ */
int deltr;                  /* R信号誤差    */
int deltg;                  /* G信号誤差    */
int deltb;                  /* B信号誤差    */
int pnumber;    /* 配列アドレス   */
int minerr;                 /* 最小トータル誤差 */
int err;                    /* トータル誤差     */
/*  パレット番号
    0x000000 黒  0x0f0000  青   0x00000f 赤  0x0f000f 紫
    0x000f00 緑  0x0f0f00  水色 0x000f0f 黄  0x0f0f0f 白
    0x010101 暗灰 0x080000 暗青 0x000008 暗赤 0x080008 暗紫
    0x000800 暗緑 0x080800 暗水色 0x000808 暗黄 0x080808 暗白
*/
        /* 基準データ */
        refr[0] = 0;refg[0] = 0;refb[0] = 0;
        refr[1] = 0;refg[1] = 0;refb[1] = 0;
        refr[2] = 0;refg[2] = 0;refb[2] = 0;
        refr[3] = 0;refg[3] = 0;refb[3] = 0;
        refr[4] = 0;refg[4] = 0;refb[4] = 0;
        refr[5] = 0;refg[5] = 0;refb[5] = 0;
        refr[6] = 0;refg[6] = 0;refb[6] = 0;
        refr[7] = 0;refg[7] = 0;refb[7] = 0;
        refr[8] = 0;refg[8] = 0;refb[8] = 0;
        refr[9] = 0;refg[9] = 0;refb[9] = 0;
        refr[10]= 0;refg[10]= 0;refb[10]= 0;
        refr[11]= 0;refg[11]= 0;refb[11]= 0;
        refr[12]= 0;refg[12]= 0;refb[12]= 0;
        refr[13]= 0;refg[13]= 0;refb[13]= 0;
        refr[14]= 0;refg[14]= 0;refb[14]= 0;
        refr[15]= 0;refg[15]= 0;refb[15]= 0;

        for(b=0;b<16;b++){
                for(g=0;g<16;g++){
                        for(r=0;r<16;r++){
                                pnumber=256*b+16*g+r;
                                minerr=48;
                                j=0;
                                for(i=0;i<16;i++){
                                        deltr=abs(refr[i]-r);
                                        deltg=abs(refg[i]-g);
                                        deltb=abs(refb[i]-b);
                                        err=deltr+deltg+deltb;
                                        if(minerr>err){
                                                minerr=err;
                                                j=i;
                                        }
                                }
                                table[pnumber]=j;
                        }
                }
        }
}

/*
**************************************
* パレット番号変換テーブル作成関数Ⅱ *
**************************************
*/
void mktable2(table,rr,gg,bb)
unsigned char table[]; /* パレット番号変換テーブル */
unsigned char rr[];             /* パレットデータR   */
unsigned char gg[];             /* パレットデータG   */
unsigned char bb[];             /* パレットデータB   */
{
static unsigned char refr[16]; /* 基準信号R */
static unsigned char refg[16]; /* 基準信号G */
static unsigned char refb[16]; /* 基準信号B */
int r,g,b;              /* ループパラメタ */
int i,j,k;              /* ループパラメタ */
int deltr;              /* R信号誤差     */
int deltg;              /* G信号誤差     */
int deltb;              /* B信号誤差     */
int pnumber;    /* 配列アドレス   */
int minerr;             /* 最小トータル誤差 */
int err;                /* トータル誤差     */
        /* 基準データ */
        for(i=0;i<16;i++){
                refr[i]=floor((double)rr[i]/16.0+0.5);
                refg[i]=floor((double)gg[i]/16.0+0.5);
                refb[i]=floor((double)bb[i]/16.0+0.5);
        }

        for(b=0;b<16;b++){
                for(g=0;g<16;g++){
                        for(r=0;r<16;r++){
                                pnumber=256*b+16*g+r;
                                minerr=48;
                                j=0;
                                for(i=0;i<16;i++){
                                        deltr=abs(refr[i]-r);
                                        deltg=abs(refg[i]-g);
                                        deltb=abs(refb[i]-b);
                                        err=deltr+deltg+deltb;
                                        if(minerr>err){
                                                minerr=err;
                                                j=i;
                                        }
                                }
                                table[pnumber]=j;
                        }
                }
        }
}
```

【リスト3】CRTDSP.C おわり

各種の音声ラインに対応する
ラインレベル・コンバータの製作

●大塚 明●

アマチュア用として作られた民生機が年々高性能化し，多くの機種がプロの現場にも進出しています。また逆に，「業務用」とされる機器にも比較的低価格の製品が現われ，アマチュアのハイエンド・ユーザーに使われています。これは音響・映像の，さまざまな分野で起きている現象です。

それだけ「民生機」と「業務用機」の性能差が小さくなっているわけで，とても結構なことなのですが，いくつか困った問題も生じています。例えば音声信号では，〝ラインレベル〟の違いと〝伝送モード〟の違いが，時として悩みのタネになります。

そこで，現在一般に採用されている各種のラインレベルを自由に変換し，アンバランス↔バランスの伝送モードも変換してしまうアダプタを作ってみました(**写真1**)。市販品には「民生機レベル→業務用機レベル」のコンバータはありますが，本機のような，いわば全レベル対応の機種はありません。録音PAの現場では，かなり重宝に使ってもらえると思います。

〈写真1〉
ラインレベル・コンバータの外観

レベル	実効値(Vrms)	伝送モード	用途／備考
+4dBV	1.585	――――	通常は使われない
+4dBm	1.228	バランス	業務用標準
0dBV	1.0000	バランス	一部の業務用機
0dBm	0.7746	バランス	業務用機
-10dBV	0.3162	アンバランス	民生機の概略出力／電子楽器出力
-10dBm	0.2449	アンバランス	あまり使われない
-20dBV	0.1000	(バランス)	通常は使われない／まれに業務用機
-20dBm	0.0775	(バランス) アンバランス	通常は使われない／まれに業務用機 電気楽器の出力にも、どうにか対応

【表1】各種のラインレベル

「ライン」のいろいろ

ラインレベルとは，ご存知のとおり，そのシステムでの標準的な信号レベルのことです。民生機では通常，－10dBV(約310mVrms)程度であり，信号の伝送形態は単芯シールド線を用いたアンバランスです。一方，プロ機では2芯シールド線によるバランス伝送が主流で，信号レベルも0dBm～＋4dBm(約0.775～1.23Vrms)と比較的大きくなっています。これら各種のラインレベルを**表1**にまとめます。

録音・PAに使われるプロ用のミキサ卓は，どんな信号レベル・伝送モードにも対応していますから，全部の信号をいったんミキサ卓に立ち上げて，卓から各機材に分配するのなら，あえてレベル・コンバータなど必要はありません。しかし，実際には機材どうしを直接つないだほうが早い場合もありますし，ミキサ卓の入力チャネルにも限りがあります。

そのため現場では，レベルが違っても「目をつぶって」つないだり，バランス入力端子にアンバランス信号を，不利を承知で(S/Nが悪化する)入力するなど，不合理な結線も行われています。

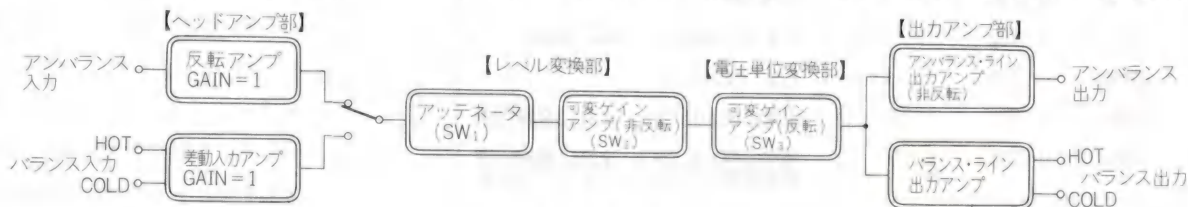
エンジニア諸氏も，そんな状態に満足してはいないはずです。本機も某スタジオ・オーナーとの雑談から生まれました。「きちんとレベル変換できて，バランス／アンバランスも換えられる〝箱〟があれば便利だね」といったオーナー氏の言葉に「そんなの簡単に作れるよ」と私が答えたのが，本機の始まりでした。

信号電圧の単位

製作には直接の関係はありませんが，ラインレベルを表す信号電圧の単位について触れておきましょう。表1でもわかるように，通常，ラインレベルにはdB*という電圧単位が用いられます。この場合の〝dB〟は，デシベルではなくディービーと読まれるのが普通です。しかし，これがまた多種多様で，混乱の元になっています。

本来〝dB〟は比を表すものです。これで絶対電圧を表現するには，0dBの電圧を決め，それとの比で表します。例えば，1Vを0dBとすれば，2倍の2Vは＋6dB，10倍の10Vは＋20dBになります。

単位	0dB(Vrms)	備考
dBV	1.000	負荷の規定は無し
dBs	1.000	dBVと同じ
dBm	0.7746	負荷を600Ωに規定
dBv	0.7746	負荷の規定は無し
dBu	0.7746	dBvと同じ

【表2】各種の信号電圧単位(一般例)

逆に1/10の0.1Vは－20dBです。

この0dBを何Vにするかで，いろいろなdB*が存在します。上の例の「0dB＝1Vrms」は，最近かなり普及している単位で〝dBV〟というものです。

プロの世界では〝dBm〟が広く使われてきました。これもたしかに電圧を表す単位なのですが，ちょっと特殊で，電力伝送の考え方が基礎になっています。すなわち，「600Ωの負荷に1mWの電力を消費させる電圧」を0dBと規定しているのです。昔のプロ機器では，入出力に600Ωのトランスが入っていたので，こういう規定のほうが便利だったのかもしれません。具体的な電圧は約0.7746Vrmsです。

現在，ほとんどの機材がトランスレスになっています。そして，送り出しインピーダンスを100Ω程度と低くし，受け側の入力インピーダンスを数kΩ以上にする，いわゆる「ロー出し・ハイ受け」の電圧伝送が主流になっています。こうなると，電力伝送を基礎に置いたdBmは，あまり実用的ではなくなってしまいました。

とはいえ，昔の機材との共用性からもdBmを一挙に廃止してしまうのも困ります。そこで，出てきたのが約0.7746Vrmsを規準の0dBとしながらも，負荷条件を取り払った単位〝dBv〟です。vが小文字であることにご注意ください。

このほかにも，カタログなどで使われている単位がいくつかありますので，**表2**にまとめました。つまり，いろんな単位があっても，結局は，規準電圧は0.7746Vと1Vの2種類なのです。本機では，変換の対象をこの2種類にしています。

なお，この記事では，それぞれをdBmとdBVで表現しました。本当はdBmではなくdBvなのですが，大文字と小文字では間違いやすく，また，これまでの慣用も考慮してあえてdBmにしました。

付記すれば，このdB*の表現方法はメーカーによってさまざまですから，表2は全世界的に確定的なものではありません。例えばカタログ・スペックで，dBVを使いながら，末尾に「ただし，0dBV＝0.775Vrms」となっている例もあり，これも反則とは言えません。また，単位にただの〝dB〟だけを記し，末尾などに「0dB＝1Vrms」などとしてあってもいいのです。

変換対象のレベルと全体の構成

本機の扱う信号レベルは表1の，－20，－10，0，＋4の各dBmとdBVで，伝送モードはそれぞれのバランスとアンバランスです。例えば，民生用カセットデッキの「－10dBVアンバランス」を，業務用機器の「＋4dBmバランス」に，確実に変換します。回路は独立2チャネルですからステレオでも使え

【図1】ブロック図

ます。

全体のブロックは**図1**です。ヘッドアンプはアンバランス用とバランス用があり，どちらもゲインは1（0dB）です。どちらも反転アンプで，後段の〝電圧単位変換部〟で，もう一度位相反転しています。

アンバランス／バランスの，どちらのヘッドアンプを生かすかは，その出力をスイッチで切り替えます。本機ではスイッチに，アンバランス入力ジャックのスイッチ部を使い，ここにプラグがささるとアンバランスが生き，プラグがささっていないとバランスが生きるようにしました。もちろん，通常のトグルスイッチなどに置き換えても構いません。

スイッチを経た信号は，SW_1の〝アッテネータ〟に入ります。アッテネータは，信号を「下げる」方向でレベル変換する場合に使います。減衰は6ステップで，0.063／0.1／0.199／0.316／0.631／1です。

レベルを「上げる」変換は，次の〝可変ゲインアンプ〟で行います。ゲインはオペアンプの帰還率で変え，そのスイッチはSW_2です。ここの増幅率は，1／1.585／3.16／5.016／10.0／15.85の，やはり6種類になります。アッテネータの減衰率とともに，これらの数字を**表3**の「レベル変換ゲイン」の項に示します。

ここまでの回路での減衰・増強では，信号レベルの単位は考慮されていません。単に定率でレベル変化させていたわけです。単位の変換は〝電圧単位変換部〟で行います。やはり，オペアンプの帰還率を変えて，電圧単位によるレベルの差を補正しています。たった2.2dBですが，その数字を表3の「電圧単位の変換」に示します。

以上で「レベル変換」はできました。その出力はまず非反転の出力アンプを経て〝アンバランス出力〟に，バランス・ライン出力アンプを経て〝バランス出力〟になります。ですから，この両出力は常に両方とも生きており，相互に干渉することなく，別ラインとして扱えます。

実際の回路とパーツ

片チャネルの信号回路と電源部が**図2**です。図1と見比べてもらえば概略はおわかりでしょう。

本来，このように精密なレベル変換を行う回路では，パーツも高精度の製品を使うのが常道で，抵抗も1％物を指定するのが普通です。しかし今回，私は全部の抵抗を5％物で済ませました。一つには，最近の抵抗は5％物であっても誤差は非常に小さく，選別すれば誤差1％以下のものがたくさん得られること。もう一つには，ラインレベルの精度自体が，それほどシビアではなく，最大5％程度違っても大勢に影響はないからです。

5％物のE24系列を使うため，アッテネータ部，レベル変換の帰還抵抗には，かなり面倒な計算を要しました。この定数で，変換誤差は1％以下になるはずです。

もしも1％抵抗を使うのなら，差動入力ヘッドアンプの4本の24kΩと，バランス出力回路の2本の100kΩ（どちらも片チャネル分の本数）に採用すれば有意義でしょう。これらはバランス信号の「ホット」と「コールド」のレベル差を決める部分であり，信号のひずみ率に関係するからです。

オペアンプにはJRCの4580Dと5532DDを使いました。音質を考えれば全部5532にしたいところですが，このICは入力インピーダンスが低く，差動入力範囲も小さい欠点があります。そこで，高入力インピーダンスが求められる個所や，大きな差動入力が加わる可能性のあるところには4580Dを採用しました。といって，これもかなり優秀なICですから，全体の音質は十分にプロの水準です。

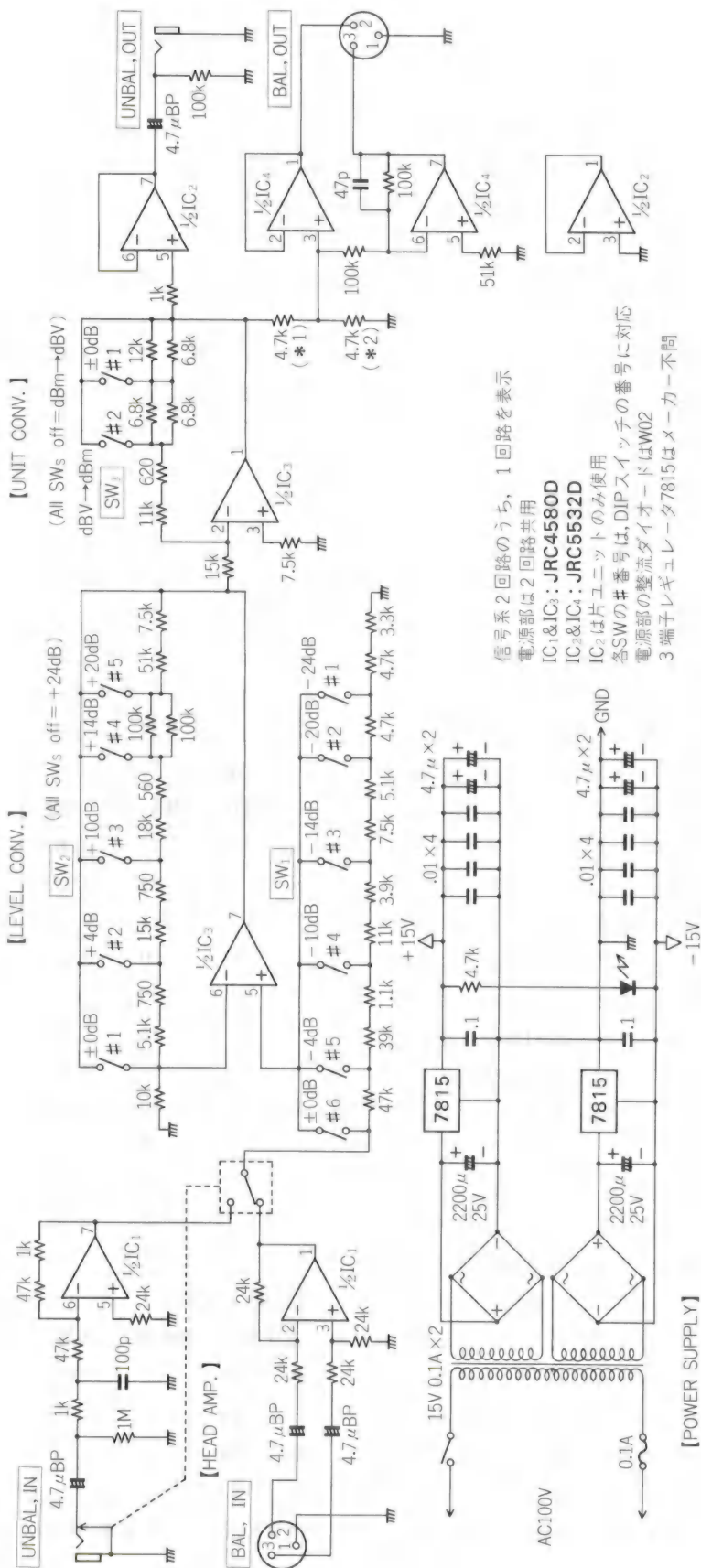
IC_2の5532は片側のみ使用し，片側は遊ばせています。これは2チャネルを完全に分けたかったからです。本機のアンバランス出力を使わないのであれば，IC_2は抜いて

［レベル変換ゲイン］

レベル変換(dB)	増減(dB)	実数
+4 → -20	-24	0.0631
+4 → -10	-14	0.199
+4 → 0	-4	0.631
0 → -20	-20	0.1
0 → -10	-10	0.316
0 → +4	+4	1.585
-10 → -20	-10	0.316
-10 → 0	+10	3.16
-10 → +4	+14	5.016
-20 → -10	+10	3.16
-20 → 0	+20	10.0
-20 → +4	+24	15.85

［電圧単位の変換］

電圧単位	増減(dB)	実数
dBm → dBV	+2.22	1.291
dBV → dBm	-2.22	0.7746
dBm → dBm	0	1.0
dBV → dBV	0	1.0

【表3】信号レベル変換の増幅・減衰率

【図2】ライン・レベル・コンバータ回路図

しまっても構いませんし，この出力に10kΩ以下の負荷を接続しないのなら，5532ではなく4558など，もっと安価なICに差し替えてもいいでしょう。

回路定数で問題になりそうな個所としては，アンバランス→バランス変換部の前に設けた，2本の4.7kΩによる分圧器があります。ここで信号を半分にしているのは，バランス出力の「ホット」「コールド」双方の振幅を1/2にするためです。ここで1/2にしないと，バランス出力を差動受けした場合，振幅は規定レベルの2倍になってしまいます。

しかし，分圧器の下側の4.7kΩ(＊2の抵抗)にはコールド出力用アンプの入力抵抗100kΩがパラになるため，合成抵抗値は4.49kΩになって，1/2の分圧比はくずれてしまいます。結果として，バランス出力はアンバランス出力に対して1％程度小さくなります。私は，これを誤差範囲と見なしていますが，気になる方は＊2の4.7kΩを4.94kΩ(240Ωを直列に入れる)にすれば，この誤差は解消するはずです。

バランス出力の反転アンプには，帰還抵抗とパラレルに47pを入れられるようにしてあります。もしもコールド出力が高域で発振したら入れてください。発振しなければ不要です。なお本機では，いわゆるスタジオ仕様に合わせて，キャノン・コネクタの接続を〝2番ホット〟としています。3番ホットのシステムでは，4個のキャノンの2番と3番の結線をそれぞれ入れ替えればOKです。

本機の各レベル変換スイッチにはDIPスイッチを用いました。これ

はプロトタイプのつもりで作ったからで，使用状況に応じてロータリ・スイッチなどに置き換えることも可能です(後述)。

電源部は2チャネル共通です。電圧安定化は3端子レギュレータで簡単に済ませています。ただし，整流用のダイオード・ブリッジは2個使い，マイナス側にも正電圧用の7815を用いました。正負に同じロットの7815を使えば，トラッキングはほぼ完璧です。

プリント基板とケース

抵抗の数が多いため，かなり大きな基板になってしまいました。サイズは85×135ミリです。電源部を含めた全回路が載り，プリントパターンを**図3**に示します(原寸)。**写真2**が組み上がった基板です。

プリントパターンには，いわゆる「ピン間通し」の線はありませんから全部手描きでいけます。配線の引き出し穴は2.54ミリ間隔で，基板コネクタも使えます。

DIPスイッチの底面に不要と思える穴がありますが，これらは全部アースです。スイッチを外付けする際のシールド線のアミ線用です。DIPスイッチしか使わない方は，これらの穴はあけなくても構いません。

パーツ面には，できれば写真のように，チャネルの区切りの線とスイッチの番号を描いておいたほうが便利でしょう。DIPスイッチ仕様では，基板が「操作部」になりますから，これは不可欠です。

各配線引き出し穴の意味と接続先を**表4**にまとめます。

ケースにはタカチのCU-3を採用しました。あるいは，もう少し小型のケースでもいいかもしれません。しかし，後述する「改造」の可能性も考慮すれば，CU-3が最適でしょう。

私の場合の穴あけ図面を**図4**に示します。**写真3，4**がフロントパネルとリアパネルです。注意すべき個所はキャノンのビス穴です。事前にいくら慎重に位置を決めても，取り付ける段になると必ずズレるから不思議です。最良の方法は，まず中央のコネクタ本体用の大穴をあけ，原物合わせでビス穴位置を決める。これしかありません。なお，キャノン・コネクタの取り付けには皿ビスを用いるのが

〈写真2〉パーツの取り付けを終えた基板

【図3】プリントパターン(原寸大)

[信号系　R·Lとも]

番号	入出力信号	接続先	配線材
1	アンバランス・ヘッドアンプ出力	入力ジャックSW部	単芯シールド線
2	1のGND		
3	アンバランス・ヘッドアンプ入力	入力ジャック	単芯シールド線
4	3のGND		
5	バランス・ヘッドアンプ出力	入力ジャックSW部	単芯シールド線
6	5のGND		
7	バランス・ヘッドアンプ，ホット入力	入力キャノン・コネクタ2番	2芯シールド線 or 三つ編線
8	7&9のGND	入力キャノン・コネクタ1番	
9	バランス・ヘッドアンプ，コールド入力	入力キャノン・コネクタ3番	
10	アンバランス・ライン出力	出力ジャック	単芯シールド線
11	10のGND		
12	13のGND		単芯シールド線
13	アッテネータ(レベル変換部)入力	入力ジャックSW部	
14	バランス・ライン，ホット出力	出力キャノン・コネクタ2番	2芯シールド線 or 三つ編線
15	14&16のGND	出力キャノン・コネクタ1番	
16	バランス・ライン，コールド出力	出力キャノン・コネクタ3番	

[電源部]

記号	接続先	配線材
イ	電源トランス巻線#1へ	ヨリ線
ロ	電源トランス巻線#1へ	
ハ	電源トランス巻線#2へ	ヨリ線
ニ	電源トランス巻線#2へ	
ホ	LEDアノードへ	ヨリ線
ヘ	LEDカソードへ	

【表4】基板穴の入出力信号と接続先

普通です。

電源オン表示のLEDの位置も，ちょっと微妙です。仕上がりでINとOUTのコネクタ間の中央にくるように位置を決めます。図面には一応の寸法を記入してありますが，あまりアテにはしないでください。まあLED位置が左右に片寄ったところで，ちょっと格好が悪くなるだけですが。

リアパネルには，とりえあず電源関係の穴が3個あくだけです。

結　線

配線材には，電源関係を除いて，単芯シールド線，2芯シールド線(または3つ編線)を使います。電源トランス内蔵ですから，ここは安全策をとりましょう。

表4だけでも配線はできますが，一応，**図5**に結線図を載せておきました。キャノン・コネクタへの配線は2芯シールド線か三つ編線で行います。三つ編線もきつく編めば相当に効果があります。線材の色も自由に選べ，シールド線よりも安価です。

そのほかの信号線には単芯シールド線を使います。シールドのアミ線は，基板側でのみハンダ付けし，配線先では切り落とし，熱収縮チューブで絶縁して無接続とします(図中★印)。これはアース・ループを作らないための措置です。ただし，Rチャネルのアンバランス入力のシールド線だけは，アミ線をジャックでアースに落とし，アース・ポイントにしています。

LEDとトランスへの配線にはより線を使います。LEDのリード線部分には熱収縮チューブを被せ，しっかりと絶縁してください。トランスへの配線は，ここには交流が流れますから，なるべくケースに這わせるように引き回します(**写真5**)。あまりブラブラさせておくと，ハム・ノイズ発生のアンテナになってしまいます。

使い方

ただ信号レベルと伝送モードを変換するだけですから，使い方はいたって簡単です。まず，あらかじめ「どんな変換を行うか」をDIPスイッチでセットします。一番S/Nの良い設定を**表5**にまとめました。この表をコピーして，ケース上ぶたの裏にでもはっておけば便

〈写真３〉フロントパネル

〈写真４〉リアパネル

利かもしれません。

例えば，「－10dBVのアンバランスから＋4dBmのバランス」の変換であれば，SW_1は６番をオン，SW_2は４番をオン，SW_3は２番をオンにセットします。そして，入力をフォン・ジャックに入れ，キャノン・コネクタから出力を得ます。この設定はチャネルごとに独立して行えますから，２チャネルをまったく別のゲイン変換にセットしても使えます。

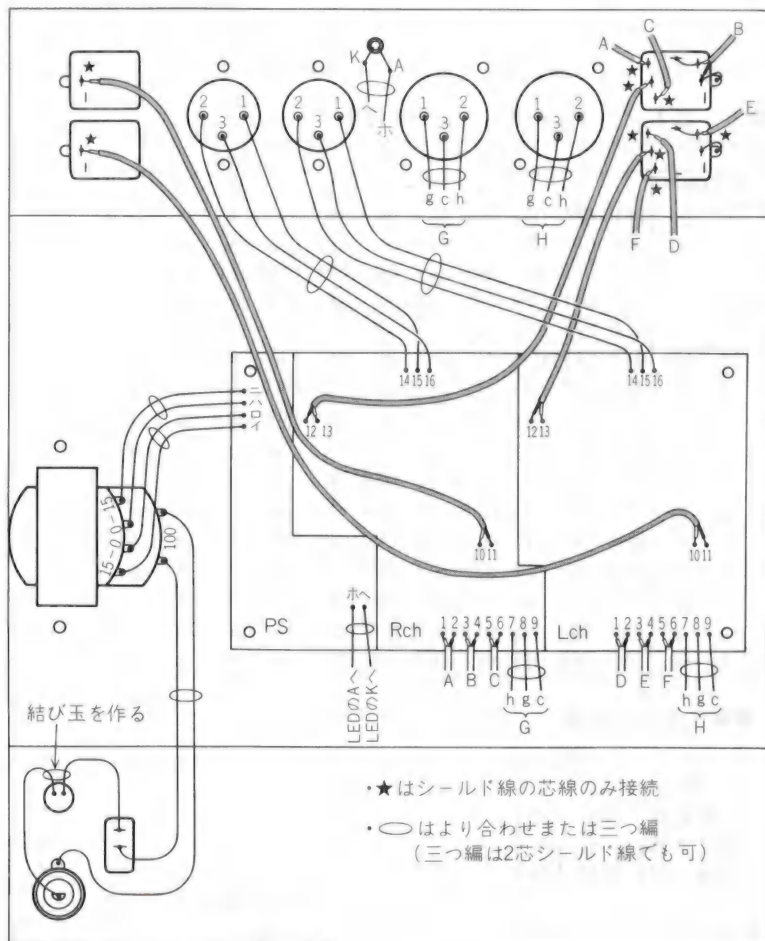
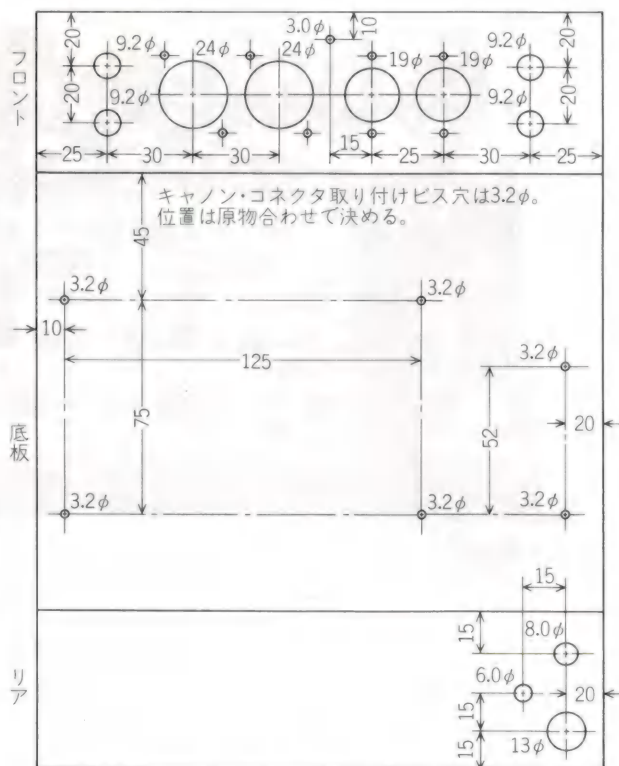
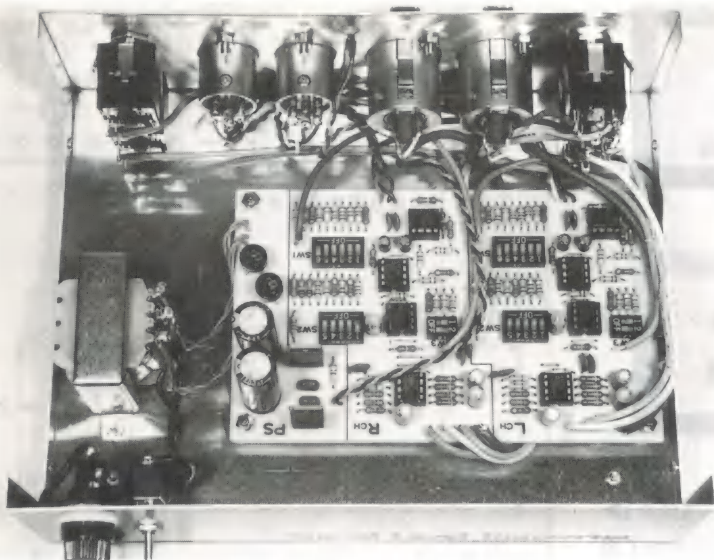
ただし同一チャネルで，フォン・ジャックとキャノン・コネクタの両方に入力しても，フォン・ジャックが優先し，キャノンに入れたバランス信号は不使用になります。

本機は「ラインレベル変換」が主目的ですが，最大で＋26dB以上のゲインがあるため(－20dBm→＋4dBVの設定時)，ブースタやゲインを持ったダイレクト・ボックスとしても使えます。また，バランスとアンバランスの出力を同時に得られることから，信号分配器にもなります。このほかにもさまざまな応用が可能でしょう。

考えられる改造

●スイッチをパネル付けにする

【図４】▶
ケース加工図
(CU-3の場合)

【図５】
▼結線図

〈写真5〉配線を終えた内部

変換レベルの設定をDIPスイッチにしたことは，便利でもあり不便でもあります。本機をある程度固定した条件で使う場合，ゲイン設定がそう簡単に変わっては困りますから，DIPスイッチが適しています。しかし，個人のレコーディング・システムなどで，時に応じていろいろなラインに本機を使うなら，ゲイン設定は簡単に変えられないと不便です。

そこで，多少強引ですが，DIPスイッチをロータリ・スイッチに置き換えてパネルに出すこともできます。リアパネルで現在不使用になっているスペースは，そのためのものです。使うロータリ・スイッチはSW_1・SW_2とも6接点の製品です。SW_3は3接点の製品か，センター・オフの6Pトグルスイッチにします。接続は**図6**の要領で，配線材にはすべて単芯シールド線を用いてください。

パネル面積の関係で，ここには3個しかロータリ・スイッチが付きませんから，切り替えはR・L両チャネル同時に行うことになり，チャネルごとの個別の設定はできなくなります。頑張って6個のスイッチを付ければ可能ですが……。

●アンバランス／バランスの入力選択スイッチを独立させる

現在，入力は「アンバランス優先」になっています。これはジャックのスイッチ部を使って選択しているからです。もちろんこれも独立させることは可能です。3Pのトグルスイッチをどこかに取り付け，基板の1，5，13番からジャックに行っている線を，そのトグル

●入・出力レベルとDIPスイッチの設定

		SW1						SW2				
IN	OUT	1	2	3	4	5	6	1	2	3	4	5
-20	-20	OFF	OFF	OFF	OFF	OFF	ON	ON	*	*	*	*
-20	-10	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	*	*
-20	0	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	ON
-20	+4	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF
-10	-20	OFF	OFF	OFF	ON	OFF	OFF	ON	*	*	*	*
-10	-10	OFF	OFF	OFF	OFF	OFF	ON	ON	*	*	*	*
-10	0	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	*	*
-10	+4	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON	*
0	-20	OFF	ON	OFF	OFF	OFF	OFF	ON	*	*	*	*
0	-10	OFF	OFF	OFF	ON	OFF	OFF	ON	*	*	*	*
0	0	OFF	OFF	OFF	OFF	OFF	ON	ON	*	*	*	*
0	+4	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	*	*	*
+4	-20	ON	OFF	OFF	OFF	OFF	OFF	ON	*	*	*	*
+4	-10	OFF	OFF	ON	OFF	OFF	OFF	ON	*	*	*	*
+4	0	OFF	ON	OFF	OFF	OFF	OFF	ON	*	*	*	*
+4	+4	OFF	OFF	OFF	OFF	OFF	ON	ON	*	*	*	*

●電圧単位の変換

	SW3	
単位	1	2
無変換	ON	OFF
dBV→dBm	*	ON
dBm→dBV	OFF	OFF

*:ON/OFFどちらでも可

【表5】
レベル変換DIPスイッチ設定一覧

【図6】変換レベル設定SWを外付けにする

スイッチに結線します。

●アンバランス出力を2倍のレベルにする

バランス・ラインのレベルとは「ホット」と「コールド」の線間のレベルです。そのため本機では，アンバランス→バランス変換部に，信号をわざわざ半分にして加えています。レベル変換としては正当なのですが，見ようによっては6dBのロスでもあります。

本機出力を受ける機器側で「6dB上がって来ている」ことを承知(？)なら，この損失を無くすことも可能です。基板上の＊1の4.7kΩをジャンパに置き換え，＊2を10kΩにすればOKです。

●ヘッドルームを拡げる

回路の最大出力は電源電圧で決まります。本機の場合，電源は±15Vですから，オペアンプの最大出力電圧は±13.5V程度です。しかしこれはピーク値で，実効値に直せば約9.5Vrms，さらにこれをdBmに換算すると約21.8dBmです。

この数字からわかるように，本機の出力は業務用標準の＋4dBmに対して＋18dB弱(約7.8倍)のヘッドルームしかありません。これで十分かどうかは意見の分かれるところでしょう。

電源電圧を上げれば最大出力は増大します。5532は±22V，4580は±18Vが最大定格ですから，本機では±18Vまでは上げられます。すると最大出力は±16.5V程度になり，ヘッドルームは1.5dBほど拡がります。

具体的には，電源トランスを16～18V200mA×2の製品と取り替え，平滑コンデンサ2,200μを35V耐圧のものにして，3端子レギュレータを7818にします。これで電源電圧は±18Vになります。

現在，回路の消費電流は，±15V電源で，正負とも約68mAです。この電流なら100mA×2のトランスでなんとか供給できますが，電圧を上昇させると電流は増加します。電源電圧を上げる場合には，トランスの電流容量を大きくするほうが安全です。そのため，200mA×2と指定しました。

いろいろなレベルのラインが混在し，それが整理されないまま「まあいいや」で接続されている現状から，軽い気持ちで作ったのが本機です。手軽に作ろうと思ってはいたものの，結果はかなり大きな〝箱〟になってしまいました。

使ってみると，やはりゲイン設定がやや面倒です。スイッチをオン抵抗の非常に低い電子スイッチにするか，レベル変換回路自体にVCAを採用して，設定操作系をロジックにすれば問題は完全に解決しますが，そうすると「一体何を作っているのか」わからなくなりそうです。そんな回路は私の妄想の中だけに留めておきましょう。

最後に，主要パーツリストを**表6**にまとめます。本誌発行時点で，パーツの合計価格は1万円程度です。

【表6】
主要パーツリスト

IC　JRC4580D×4
　JRC5532D×4
　7815×2
ダイオード　W02×2
　LED(3φ)×1
ICソケット(8P)×8
DIPスイッチ2P×2
　5P×2
　6P×2
抵抗(5％カーボン　計93本)
　560Ω×2，620Ω×2，750Ω×4，1k×6，1.1k×2，3.3k×2，3.9k×2，4.7k×9，5.1k×4，6.8k×6，7.5k×6，10k×2，11k×4，12k×2，15k×4，18k×2，24k×10，39k×2，47k×6，51k×4，100k×10，1M×2
コンデンサ
　セラミック　47p×2，100p×2，0.01μ×8
　マイラ　0.1μ×2
　無極性電解　4.7μ25V×8
　電解　4.7μ25V×4，2200μ25V×2
電源トランス　15V 0.1A2巻線×1
キャノン・タイプ・コネクタ
　XLR-3-31×2，XLR-3-32×2
フォン・ジャック
　ステレオSW付き×2，モノラル×2
ケース　タカチCU-3×1
トグルスイッチ2P×1
ヒューズ・ホルダ（ミニ）×1
ヒューズ0.1A×1
ACコード（プラグ付き）×1
ゴムブッシュ×1
スペーサ 3φ10mm×4
ビス・ナット
　3×12皿8組，3×18　4組，3×10　2組
配線材
　2芯シールド線，単芯シールド線，ビニール被覆線

ディジタルオーディオ
最大値測定装置の製作

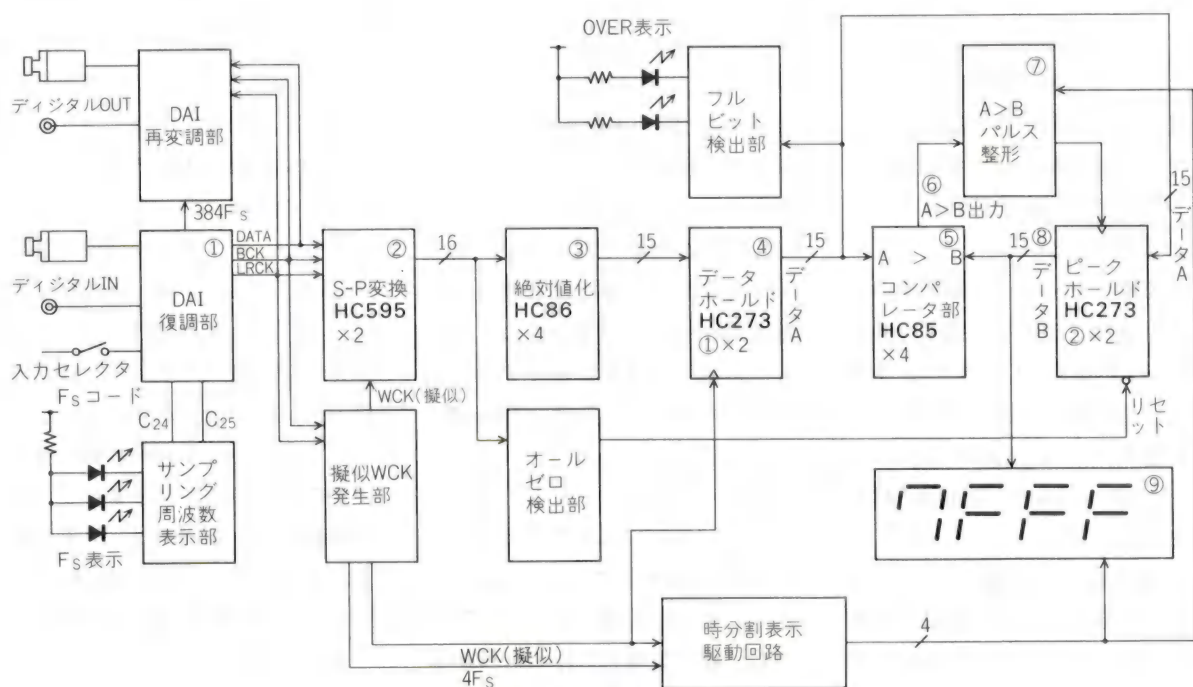
●伊藤 弘之●

今や，ディジタルオーディオもすっかり身近な存在となりました。CDやDATなどの機器は，新しいモデルが登場するごとに性能がアップし，手軽に高品質なサウンドを楽しむことができます。そんな中で，ただ聴いたり録音したりするだけではもったいないのが，このディジタルオーディオです。

皆さんもご存知かもしれませんが，大抵のディジタルオーディオ機器には，ディジタル入出力端子が備わっており，これらの機器間ではディジタルのままダイレクトで信号のやり取りが可能です。この信号は，ディジタルオーディオインタフェース信号(以下，DAI信号と略します)と呼ばれていますが，実はこのDAI信号を専用LSIで復調することで限りない応用が可能となります。

例えば，ディジタルボリュームやDSP，D/Aコンバータなどを接続すれば，自分の好みで音を変え

〈写真1〉ディジタルオーディオ最大値測定装置の外観

〈写真2〉リアパネル

【図1】本機のブロック図

たりして聴くことができます。自作派の皆さんとしては，これを利用しない手はありません。ぜひDAI信号を復調し，活用してみることをお薦めします。

今回，この復調したデータを元に，ディジタルオーディオデータの最大値を検出し，16進数4ケタでLED表示する装置を製作してみましたので，紹介してみたいと思います(**写真1，2**)。

きっかけとなったのは，ディジタルオーディオデータの正確なレベルを確認する際，DATなどのレベルメータ(dB表示)では，いまひとつピンとこなかったためです。ハードウェア派の筆者としては，1ビット単位でレベルが表示されたほうがわかりやすく各種の製作にも便利と考えました。なお，回路を検討するにあたって，本誌'89年1月号のディジタルレベルメータを参考とさせていただきました。

本機の仕組み

図1に，本機のブロック図を示します。動作の仕組みとしては，

①インタフェースLSIによってDAI信号を復調します。

②復調されたデータをシリアル－パラレル変換し，16ビットパラレルデータを得ます。

③負の値が出現した場合，絶対値化回路で正の値に直します。

④絶対値化したのち，値を正確に固定するためDフリップフロップで，一度パラレルシフトします。

⑤絶対値化されたデータをA，先に記憶されたピークデータをBとして，ディジタルコンパレータで比較します。

⑥ここでデータAの値が，データBの値より大きい場合，A＞Bのパルスが出力されます。

⑦A＞Bのパルスは，ピーク検出が連続すると，パルスの間がつながってしまうので，波形を整形します。

⑧このパルスによって，ピークホールド用のDフリップフロップ②はデータAを出力側へパラレルシフトし，新しいピークデータBとなります。

⑨更新されたデータBは，表示部により視覚化されます。

以上おおまかに説明しましたが，本機がディジタルオーディオデータの最大値を検出できる仕組みがご理解いただけたかと思います。

インタフェース部の構成

図2に示すように，DAI信号の復

【図2】インタフェースボード

+5V
V_{CC}
100μ（電解）
0V
GND
S-P変換
BCK
DATA
HC595
HC393
HC04
MSB
LSB
絶対値化
HC86
データホールド
HC273
コンパレータ
HC273のピンNo.
ROMのピンNo.
HC85
ピークホールド
HC4078
オールゼロ検出
HC595のピンNo.
HC139へ
擬似WCK発生
HC86
47p（セラ）
1k
LRCK
HC4040
HC273のピンNo.
HC30
フルビット検出
HC02
パルス成形
HC00
470
クリア表示
擬似WCK
HC04
$\overline{LRCK}$
HC00
470p
470p（セラ）
1/2 HC74
1/2 HC423
200k
470
フルビット表示部 Lch Rch
TBP18S030N
NKR161
（上面図）
表示部
HC139
時分割表示駆動部
手動，パワーオンリセット
1S1588
47k
100
4Fs
WCK（擬似）
時限　2^{14}LRCK
$= 2^{14} \times \frac{1}{44100} = 0.37$(s)
（ただしLFs：44.1kHzのとき）
・V_{CC}→GND間には，名パッケージごとに0.1μFの積層セラコンをパスコンとして入れます。
・電源ピンは省略していますが，忘れず配線してください。
【図3】メインボード

調には，専用LSI東芝TC9245Nを使用しました。

このデバイスの特徴は，3つのサンプリング周波数およびSCMSに対応し，外付け部品が少なく安定度が高いうえ，4チャネルの入力セレクタを内蔵しています。これで，価格が700円程度ですから，大変お得です(ほかにパイオニアのPD0052もお薦めです)。

ただ，シュリンクDIPパッケージのため，そのままでは2.54ピッチの基板には実装できません。通常ではピッチ変換ソケットの使用に踏み切るところですが，サンハヤトの専用基板ICB-01Bを使いました。

ここで，お気づきの方もいらっしゃるかと思いますが，ICB-01Bは64ピン用なのに，TC9245Nは28ピンしかないので，空いたスペースがもったいないとおっしゃるかもしれません。

実は，この空きスペースにデータ変調送信用のTC9231Nを実装し，おまけに送受信両方の光モジュールも載せてインタフェースユニットとして，活用しています。このユニットは，例えばディジタルボリュームのように，復調データを加工したのち，再度変調して出力する場合には，便利だと思います。高価な変換ソケットや変換基板を節約できるし，スペース的にも有利となるでしょう。

今回は，データを加工して出力しませんからそのままスルーで伝送し，復調が正しく行われているか確認のため使用することにします。

TC9245Nは光および同軸入力でDAI信号を受信すると，内蔵PLLによって，384F_S(F_S：サンプリング周波数の略)マスタークロックをはじめ，音声データ，LRクロック(LRCK)，ビットクロック(BCK)，チャネルステータス情報を出力します。

復調部は，本機の心臓部といえますので，電源ラインには特に気を配ります。2㎜幅のハンダ吸い取り線を使うなどして太めに配線し，各パッケージにはパスコンを，電源の入口にはケミコンを入れます。

チャネルステータス情報はコピーの可否，エンファシスの有無，3つのサンプリング周波数をLEDで表示しました。

F_Sコードは，そのままでは表示できないので，専用デコーダを組みましたが，面倒な方はHC139でかまいません。

シリアル－パラレル変換部の構成

続いて，**図3**に示すメインボードです。まず，復調データはシリアルで出力されてきますから，シフトレジスタHC595でパラレルデータに変換します。ここで，問題なのがシリアル－パラレル変換(略して，S－P変換)には，パラレルデータをラッチするためのワードクロック(WCK)が必要ですが，TC9245Nにはありません。

そこで苦心の末，BCKとLRCKから，何とか擬似WCKを作り出しました。仕組みは，エッジ検出回路によってLRCKのエッジごとに短いパルスを作り，カウンタHC393をリセットし，BCKをカウントします。BCKは32F_S，WCKは2F_Sですから16分周すればよいことになります。エッジごとにリセットしないと分周はされますが，LRCKとタイミングが合いません。

カウンタは同期タイプを使用すれば精度も上がりますが，WCKは主としてデータラッチに使用されますのでLRCKとタイミング，位相が合っていれば多少の遅延は問題ないと考えたため，非同期としました(正確にはWCKとはいえないので，擬似WCKとしました)。

絶対値化回路

S－P変換後データを絶対値化します。ディジタルオーディオのレベルは，16ビットの2の補数で表現され，正の領域が0000～7FFF(10進表現で0～32767)負の領域がFFFF～8000(10進表現で－1～－32768)で，トータルで2^{16}=65536

です。

図4をご覧いただくとわかりやすいと思いますが，ディジタル値として正負の境界線は0と－1の間にあります。レベルとしては，正負の値とも絶対値をとりますから，正の値はそのままで，負の値のみ全ビット反転します。負の値はMSBが1になりますから，HC86を使って全ビットひっくり返します。

ここまでで，お気づきのとおり，MSBは正負の符号を，以下のビットがレベル値を表しますから，絶対値化したのちは，MSBは考慮しなくてもよくなります。

さて，レベル値だけをうまく取り出せましたが，このままでは，コンパレータに入れることはできません。HC86でデータを反転する際，MSBに入っているインバータによる遅延が原因となって，ほんの一瞬，全く違った値が出現します。対策として，値が十分安定したと思われる1/2WCK周期後に，一度パラレルシフトします。これには，DフリップフロップHC273を使い，シフトクロックは擬似WCKを使用しています。

コンパレータ部

安定化したデータは，16ビットディジタルコンパレータによって先に記憶されているピークデータと比較されます。ここでは，比較するデータをA，ピークデータをBと仮定します。HC85は4ビットのディジタルコンパレータで，AとBの入力を比較し，A＜B，A＝B，A＞Bの判定出力を出しますが，今回はA＞Bだけを使用します。また，本機では16ビットのデータを扱うわけですから，HC85を4つカスケード接続しています。

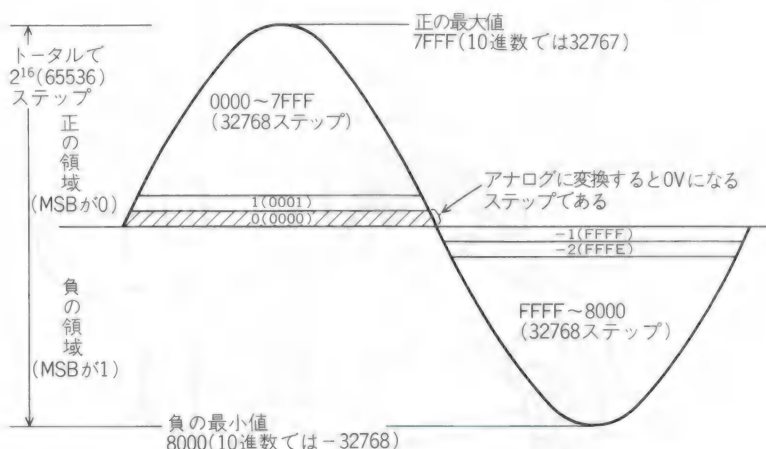
動作の仕組みは，まずピークホルド用のHC273②の出力は，パワーオンリセットによってオールゼロになっています。つまり，データBはゼロです。次に，新しく入ってきたデータAとコンパレータで比較します。もし，データAがデータBより大きい場合，最上位のHC85より，A＞Bのパルスが出力され，このパルスによってDフリップフロップ②をシフトします。

結果，データAは古いデータBに代わって，新しいデータBとなり，次の比較を待ちます。

しかし，ピーク検出が連続すると動作がおかしくなります。原因は，疑似WCK幅で出力されるA＞Bパルスが連続すると，パルスのとぎれ目がなくなり，パルス幅が広がって正確な値をホールドできないためとわかりました。そこで，HC139を使い疑似WCKと$4F_S$クロックから4相クロックを作り，NORゲートで合成したところ，パルス幅が狭まり，誤動作はなくなりました。HC139の4つの出力の内，Y_3が最も確実なようです。

表示部

表示部は，16進数で4ケタのLED表示ですが，デコーダによい物が見つかりませんでした。そこで，'90年3月号にありましたP-ROM TBP18S030Nを使って表示用デコーダを作ってみました。書き込みデータとアドレスは**表1**を，ROMライタは'90年3月号を参考にしてください（'90年3月号とはアドレスが逆ですが，3月号はアドレスの前にインバータが入っているためです）。

7セグメントLEDとデコードROMそれぞれ4つをスタティック駆動すると，かなりの電流を消費します。そこで，A＞Bパルスを整形するときに使った4相クロックで，ROMの出力イネーブルをコントロールし，4つの表示器を順次点灯しています。7セグメントLEDには，スタンレー製の超高輝度タイプNKR161を使いましたので，時分割駆動で減少した分の明るさを補うことができました。結果，消費電流は1/3程度に減少し，

※0Vも幅をもったレベル値で決して1本のラインではない

【図4】ディジタルオーディオデータのレベル値

電源の負担も軽くなりました。

フルビット検出部

'89年1月号のレベルメータにもありましたが，検出方法が多少異なっています。レベルメータでは加算器のキャリーを利用していましたが，より正確に検出するため，パラレルデータを16ビットANDし，ワンショットに送っています。本機は最大値が7FFF(つまりフルビット)になると，比較動作をやめてしまいます。これでは，ちょっとさみしいので，追加したのが本当のところです。LEDはなるべく明るいものが，警告らしくてよいでしょう。

録音レベルの高いソフトを再生すると，この表示が続出するものがあり，ド迫力です。

リセット回路

パワーオン，手動によるリセットは，もはや当たり前ですが，CDのアルバムなどでは，曲間でいちいちリセットしていたのでは，わずらわしくて仕方ありません。多くのソフトは曲間のデータがオールゼロになることを利用して，自動的にリセットがかかるよう工夫しました。

16ビットORでオールゼロを検出し，Lの出力を得ます。これにより，カウンタIC HC4040のリセットは解除され，LRCKをカウントします。HC4040はたすきがけにより，2^{14}カウントで自動的に停止し，リセット信号を出力しますが，もしカウント中にゼロ以外のデータが検出されると，瞬時に初期の状態に戻ります。これは，瞬間的なオールゼロには反応させないための対策です。

セグメント＼アドレス	未 (Q_7)	g (Q_6)	f (Q_5)	e (Q_4)	d (Q_3)	c (Q_2)	b (Q_1)	a (Q_0)	コード
0	×	×	○	○	○	○	○	○	3F
1	×	×	×	×	×	○	○	×	06
2	×	○	×	○	○	×	○	○	5B
3	×	○	×	×	○	○	○	○	4F
4	×	○	○	×	×	○	○	×	66
5	×	○	○	×	○	○	×	○	6D
6	×	○	○	○	○	○	×	○	7D
7	×	×	○	×	×	○	○	○	27
8	×	○	○	○	○	○	○	○	7F
9	×	○	○	×	○	○	○	○	6F
A	×	○	○	○	×	○	○	○	77
B	×	○	○	○	○	○	×	×	7C
C	×	×	○	○	○	×	×	○	39
D	×	○	×	○	○	○	○	×	5E
E	×	○	○	○	○	×	×	○	79
F	×	○	○	○	×	×	×	○	71

【表1】P-ROM書き込みデータ

曲中でもオールゼロはあり得ますから，こうなるとリセットが頻発し役に立ちません。また，オールゼロ検出は，絶対値化する前に行わないと，－1 (FFFF)までゼロとみなしてしまいますから，ご注意ください。

図中のクリア表示は，動作確認のため基板上に設けましたが，7セグ表示がオールゼロを示すのでなくてもかまいません。

電源部

電源は8V500mAのACアダプタから供給し，3端子で安定化しました(**図5**)。

デバイスはP-ROMを除きオールC-MOSですから，十分余裕があります。3端子は低ドロップ型LM2940CT5.0を使ってみましたが，最低5.5V入力で安定化5Vが得られるというものです。フィルタ用のケミコンは，手持ちの関係で1,000μFを使用しましたが，もう少し大きめの容量がよいでしょう。

いよいよ製作です

部品の入手は，割とポピュラーな物ばかり使用していますから，容易かと思います(**表2**)。

インタフェースLSIは，データシートも一緒に購入するとよいでしょう。

製作は，インタフェースボードから入ります。LSIを専用基板にハ

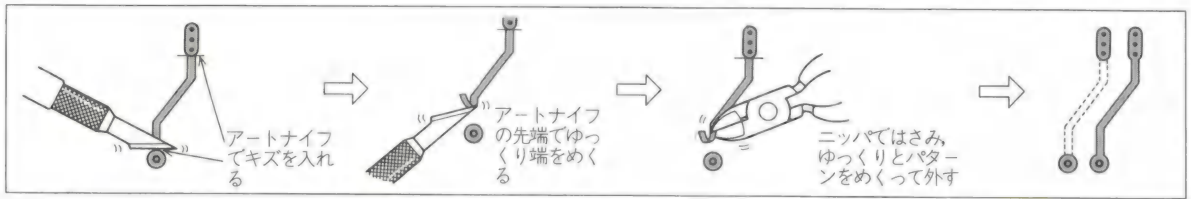
【図5】電源部

【表2】 使用部品

—インタフェース部—			
LSI	TC9245N	1	共立
	TC9231N	1	共立
IC	74HC00	1	藤商
	74HC04	2	藤商
光モジュール	TOTX172	1	共立
	TORX172	1	共立
LED	TLUR153	1	藤商
	TLUG153	1	藤商
	TLUY153	1	藤商
	TLUR164	1	藤商
	TLUG164	1	藤商
	TLUY164	1	藤商
抵抗	100kΩ	1	藤商
カーボン1/4W	47kΩ	1	藤商
	20kΩ	1	藤商
	10kΩ	1	藤商
	6.8kΩ	1	藤商
	2.2kΩ	1	藤商
	1.2kΩ	1	藤商
	750Ω	3	藤商
	470Ω	4	藤商
	330Ω	1	藤商
	82Ω	1	藤商
	75Ω	1	藤商
	10Ω	1	藤商
C電解	47μF16V	1	藤商
	33μF16V	1	藤商
	1μF25V	1	藤商
セラミック	220pF50V	1	藤商
	120pF50V	1	藤商
	22pF50V	1	藤商
ポリエステル	0.047μF50V	1	若松
積層セラミック	0.1μF50V	10	秋月
	0.01μF50V	1	秋月
マイクロインダクタ	100μH	4	秋月
基板	ICB-01B	1	藤商
ICソケット バンデイ	14P	3	藤商
ミニトグルSW	2P	1	藤商
RCAピンジャック	2P	1	イノデン
—メインボード			
IC	74HC4078	2	藤商
	HC4040	2	藤商
	HC595	2	藤商
	HC423	1	藤商
	HC393	1	藤商
	HC273	4	藤商
	HC139	1	藤商
	HC86	5	藤商
	HC85	4	藤商
	HC74	1	藤商
	HC30	2	藤商
	HC04	1	藤商
	HC02	1	藤商
	HC00	1	藤商
ICソケット バンデイ	20P	4	藤商
	16P	10	藤商
	14P	14	藤商
C 積層セラミック	0.1μF50V	31	秋月
セラミック	470pF50V	2	藤商
	47pF50V	1	藤商
電解	100μF16V	1	藤商
抵抗カーボン 1/4W	200kΩ	2	藤商
	4.7kΩ	1	藤商
	1kΩ	1	藤商
	470Ω	2	藤商
	100Ω	1	藤商
LED TLUR163		2	藤商
ダイオード	1S1588	1	藤商
基板	ICB-99L	1	藤商
2PプッシュSW		1	藤商
—表示部—			
7セグメントLED	NKR161	4	秋月
P-ROM	TBP18S030N	4	イノデン
抵抗カーボン1/4W	1kΩ	28	藤商
基板	ICB-93SG	1	藤商
ベークスペーサ	15mm	4	藤商
	5mm	4	藤商
アルミ板		少々	
—電源部—			
3端子	LM2940CT-5.0	1	秋月
C 電解	1000μF25V	1	藤商
	47μF25V	1	藤商
積層セラミック	0.1μF50V	2	秋月
ヒートシンク		1	藤商
基板	ICB-89	1	藤商
—その他—			
ケース	YM-200	1	イノデン
ラッピング線		適量	
ビニール被覆線		適量	
10連ビニール被覆線		適量	イノデン
ベークスペーサ	5mm	8	藤商
照光プッシュSW		1	イノデン
DCジャック		1	イノデン
ACアダプタ	8V500mA	1	秋月
基板コネクタ	EI -8	2	イノデン
	EI -3	1	イノデン
	EI -2	5	イノデン
	EL-5	1	イノデン
	EL-3	1	イノデン
	EL-2	3	イノデン
2mm幅ハンダ吸い取り線		適量	
ビス・ナット		適量	
インレタ		適量	

・ケース，基板はもう少し余裕のあるものが良いでしょう。
・マイクロインダクタ・積層セラミックコンデンサは10本テーピング品
・購入方法はすべて通販です。

ンダ付けし，電源ラインを配線しますが，ピッチが細かいので，ハンダブリッジに注意します。

光モジュールは，基板固定用のピンが縦横それぞれ1/2ピッチずれています。このままでは実装できませんから，基板に正確に穴位置を決め，ϕ 1 mmのドリルで穴をあけます。基板にのせたら，固定ピン

【図6】プリントパターンのはがし方

はアースラインにしっかりとハンダ付けします。これを怠ると，光ケーブルを抜き差ししているうちに，プリントパターンが剝離する恐れがあります。

各信号の配線は，ラッピングワイヤですが，384F_Sマスタクロックを伝送する線は，シールド線のほうがよいでしょう。

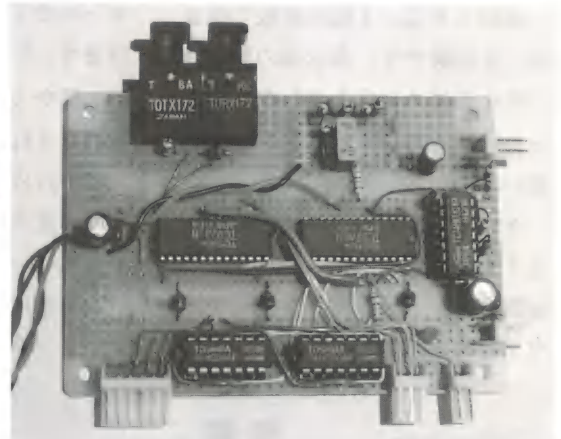
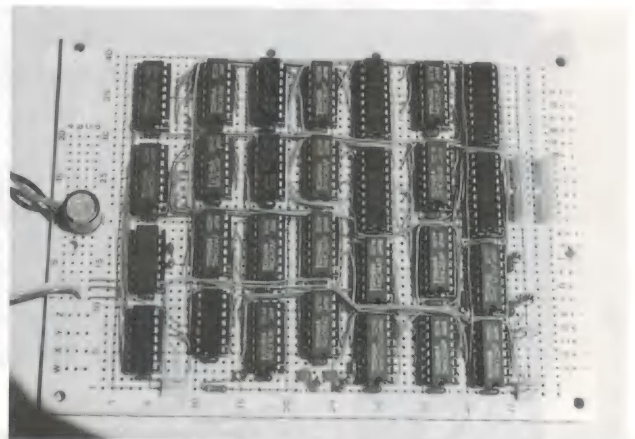
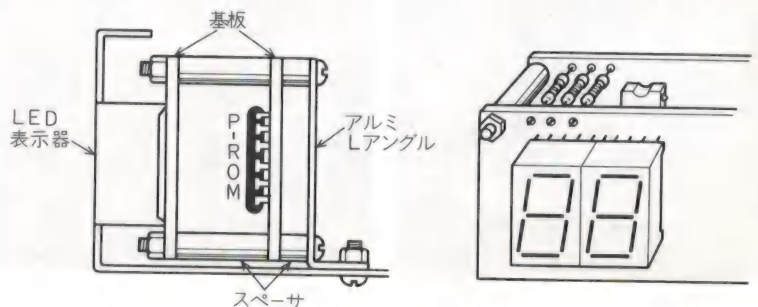
TC9245NのアナログV_{DD}は，マイクロインダクタを入れると動作が不安定になりましたので，省いています。F_S表示用デコーダのHC00，04は一部（復・変調LSIとの間）基板のプリントパターンを切除しないと実装できないかもしれません。**図6**にやり方を示します。

部品の配置は**写真3**を参考にしてください。組み上がったら入念に配線チェックを行います。

次にメインボードです（**写真4**）。配線は膨大な量ですから，気長に根気よく頑張ってください。筆者は，先にケースの大きさを決め，それから，基板を選んだので，超過密的なスペースになってしまいました。皆さんはもう少し余裕のある基板とケースにしてください。

配線はインタフェース部と同じラッピングワイヤですが，まるで，くもの巣です。各ブロックごとに結線チェックを行いながら進めてください。ICソケットの1番ピンは白のペイントマーカーで印をしておくと，確認に便利です。各ICには，忘れずにパスコンを入れておきます。すべて終わったら必ず，テスタで導通チェックをしましょう。

〈写真3〉
インタフェースボード

〈写真4〉
メインボード

表示部は，**図7**に示すとおり，立体構造です。まず，万能基板ICB-93SGを2つに切断し，1枚は7セグメントLEDを，もう1枚はP-ROMをハンダ付けします。さらにこれらを重ねて，電流制限用の抵抗器，1kΩ×28本で連結します。

【図7】表示部

ハンダ付けは7セグメント側から行い，2つの基板を抵抗器とスペーサで固定したあと，ROM側で残りをやります。順番を間違うと組み立てられませんから，ご注意ください。

面倒な方は，1枚の基板で組まれても結構です。表示部の固定は，パネル面にネジ穴を出したくなかったので，アルミのLアングルを使いました。

ケースの加工は，光モジュールと7セグメントLEDの角穴が少し面倒です。メインボードとインタフェースボードは，スペースの都合で2階建て構造になっていますから，光モジュールの角穴の位置決めに工夫がいります。

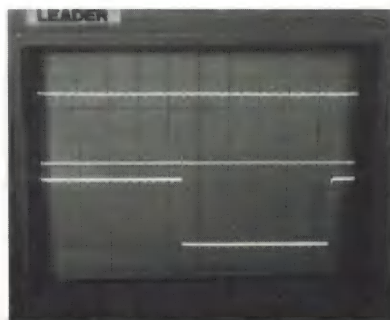
まず，2枚の基板と，スペーサを瞬間接着剤で仮に固定します。光モジュールのキャップを外し，ケースにくっ付けて鉛筆で印をつけます(スペーサは後で外しますが，横から力を加えるとぽろっと取れます)。

穴あけのとき，寸法を見ようとして，光モジュールにアルミの粉を入れないように注意してください。筆者は，用心のため受口にメンディングテープをはっておきました。

穴あけが終わったら，レタリングで文字を入れたあとに，LEDを固定します。始めに瞬間で仮固定し，さらに，5分硬化型のエポキシを使うとうまくいきます。

各基板の結線は，基板コネクタを多用しました。**写真5**は内部の様子ですが，これをハンダ付けだけで行うのは，かなり勇気がいると思います。最終チェックは，電源の極性です。これを誤ると，何10個ものICが一瞬にしてオシャカです。

〈写真5〉配線を終えた内部

〈写真6〉TC9245Nの復調波形
BCK(上)とLRCK(下) 5V/DIV

〈写真7〉TC9245Nの復調波形
DATA(上)とLRCK(下) 5V/DIV

さあ，テストです

始めは復調部からです。光または同軸入力と，CDやDATのディジタル出力をケーブルで接続します。電源を入れると，エラー表示が消灯し，F_S表示は送り出し機器のサンプリングに対応して点灯すれば，まず問題はありません。コピーの可否，エンファシスの有無はソフトにより表示が異なります。

写真6，7は，TC9245Nの復調波形です。もし，単体のDAコンバータやモニタ機能のあるDATをお持ちの方は，本機のディジタルアウトを接続し，正しく音が出るか確認してもよいでしょう。

メインボードは，まずソフトを再生しない状態で，7セグメント表示が0000を確認します。続いて曲を再生すると，順次ピーク表示が書き換えられていくと思います。さらにピーク表示が7FFFに達すると，一瞬OVER表示が点灯します。また，曲間の無音部(ノイズがある場合は，一時停止にして)でオートリセットが，手動でもリセットがかかることを確認してください。

これでほぼ完成です。

本機も一応測定装置と銘打っていますから，正確さを知りたいところです。そこで，簡単な方形波発生器を組んでみました(図8)。F_Sは48kHz，周期は3kHzで，レベルはDIPスイッチで設定します。

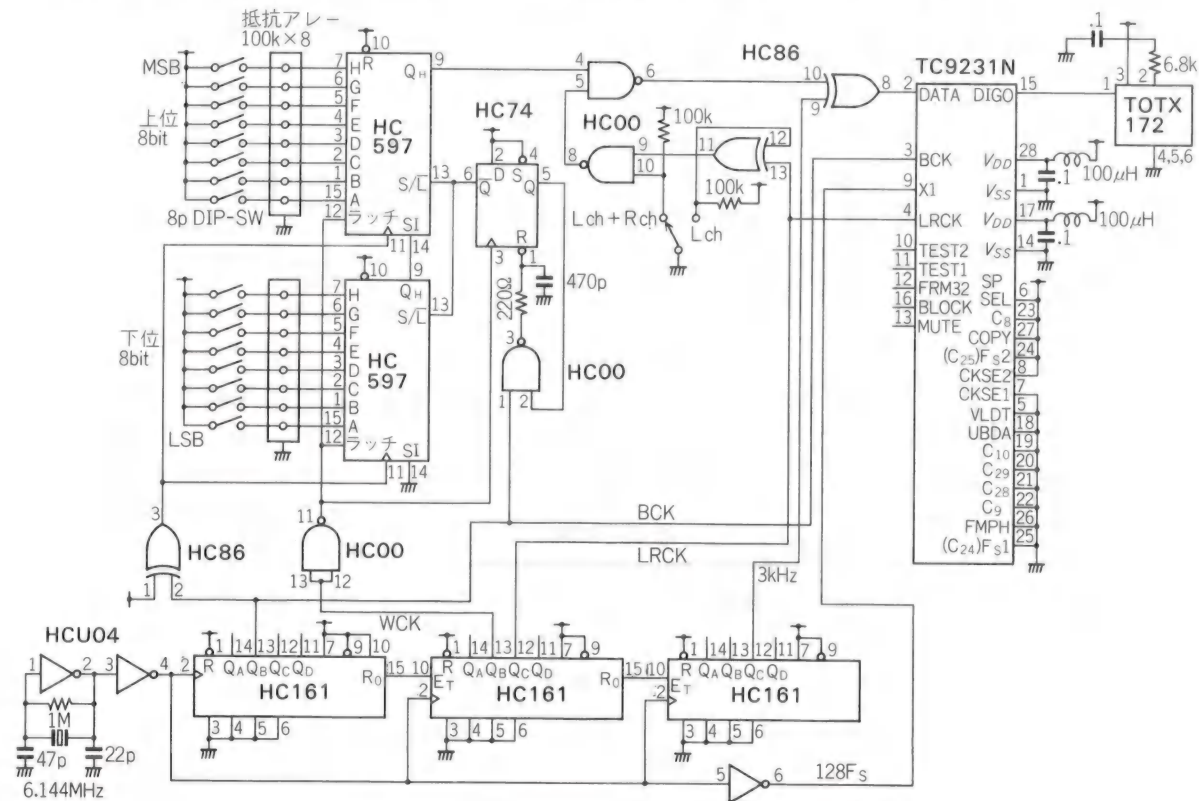
テストの方法は，まず測定装置とテスト信号発生基板を光ケーブルで接続します。次に本機を一度リセットしてから，順次テスト信号発生器側でレベルを上げていくと，それに対応して表示も変化すればOKです。

また，OVER警告表示の位相を確認するため，テスト信号発生器では，切り替えでLチャネルのみ伝送できるようにしました。これでレベルを7FFFに設定しますと，Lチャネル側のOVER表示のみが一瞬点灯するはずです。

本機は一種の電子計算機のようなものですから，結線さえ間違えなければ無調整で動作します。配線のチェックは十分過ぎるほど行ってください。

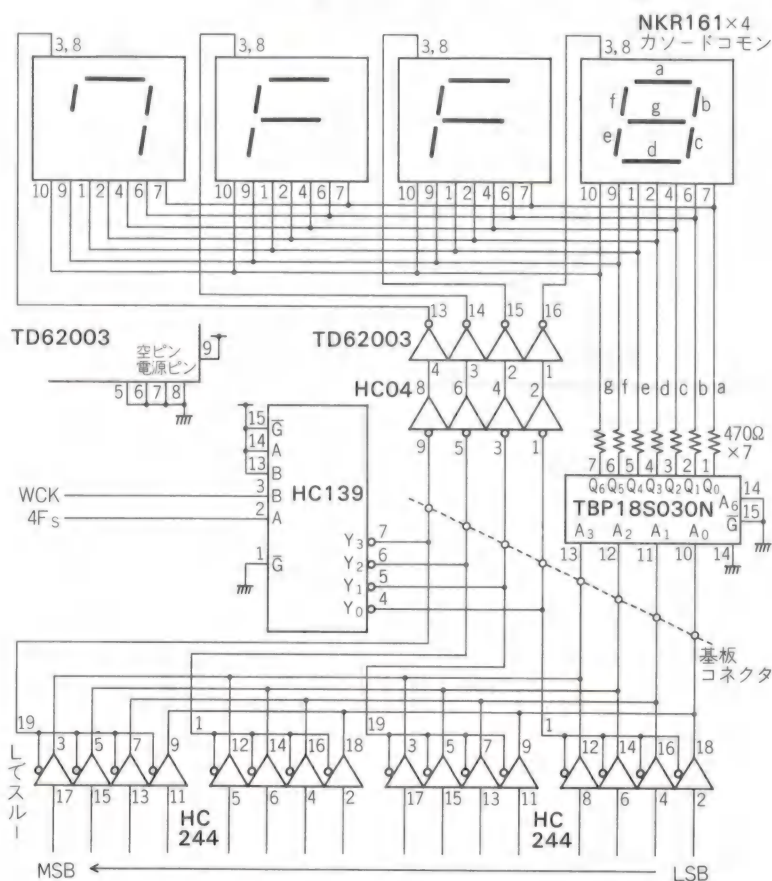
問題点

一つはオートリセットの動作で，曲間にノイズがある場合，全く反応しません。対策として，HC4078の下位4ビットくらいをLレベルに固定してみます。また，曲間の短い場合は，HC4040のたすきを変更するとよいでしょう。

表示部は，P-ROMが比較的高価でかつ電流も消費しますから，図9にROMを1個にした回路をあげておきました。

【図8】テスト信号発生器

〈写真8〉レベル測定風景

アーティスト名	曲　名	最大レベル値	備　考
長渕　剛	乾杯	7FFF	アルバム
竹内　まりや	駅	7FFD	アルバム
B'z	BLOWIN'	753E	シングル
中島　みゆき	誘惑	6A77	シングル
米米CLUB	君がいるだけで	7FFF	シングル
高橋　真梨子	はがゆい唇	7FFF	シングル
由紀　さおり	両国橋	7506	アルバム

【表3】ソフトの最大レベル測定値

【図9】改善した時分割表示駆動回路

本機は，FFFF(−1)レベルのときは，表示は0000となります。本来−1は絶対値化すると1になるのが正しいのでしょうが，データをDA変換したあとのレベルならともかく，伝送中のディジタル値を示すのですから，2^{16}を二分したデータを絶対値で表現しました。本当なら，このようなレベルの表現は，7FFF～8000の両方を示すのが適切なのかもしれません。

使用してみて

写真8に示すように本機を使用して，実際にソフトのレベルを測定してみたのが，**表3**です。普段は，このようにソフトのレベルチェックに使用していますが，ディジタルデータの振幅を直接観測できますから，信頼感があります。

ほかの活用法としては，ディジタルミキサを使用するときあらかじめソフトのレベルを測定しておけば，ディジタルボリュームによる調整も容易で，オーバーフローを未然に防止できます。そのため，筆者は'89年3月号のディジタルボリュームを，係数が上位8ビット，下位8ビットの合計16ビットで，より精密にコントロールできるよう改造しました。

今後の応用

ぜひ，正・負両方のフルスケールで測定できるものを製作してみたいと思っています。また，インタフェースLSIも，24ビットまで復調できるものも出ていますから，これを使って，将来登場が予想される20ビットディジタルオーディオに対応できる回路も考えています(このとき，フルビット表示は7FFFFとなるでしょう…)。

本機は，各段のビット数を，同じ数だけ増やしていけば，理論的に24ビットまで対応可能です。皆さんも頑張って，チャレンジしてください。

高輝度・高解像度を両立させる

ILA™スーパープロジェクタ

三好　忠義

映像情報の多様化・高画質化が進み，ハイビジョン放送規格や，CGのSVGA規格に代表される高品質画像データが増加し，これに伴って，高画質・大画面ディスプレイが不可欠になっています。特に，大画面は多人数への情報提供などの実務面から，迫力や臨場感を求める感性まで巾広い用途が考えられ，今後の映像の「核」と言えます。

従来の直視型ディスプレイも大画面化への努力を続けているものの，重量・形状・強度などより限界があり，映像を光学的に拡大投写するプロジェクタへの要求が高まってきています。

これに応える高画質プロジェクタに要求される性能は以下の通りです。

⑴イベントホール対応の大画面
　(中規模ホールで300インチ以上)
⑵映画並以上の高輝度
　(300インチで1,500ルーメン以上の光出力)
⑶ハイビジョン対応の高解像度
　(1,000TV本以上)
⑷マルチメディア対応の入力信号
　(バリアブルスキャン同期)

これらの性能を満足させるよう，新規開発の投写方式を採用したのが「ILA*スーパープロジェクタ」です。従来方式と比較しながら，ILA用のデバイス，方式，装置について紹介します。

各プロジェクタ方式の現状と課題

プロジェクタ方式として各種提案されていますが，主にCRT(ブラウン管)投写管方式とLCD(液晶)パネル方式に大別されます。**表1**に各方式の構成と特徴の概要を示します。

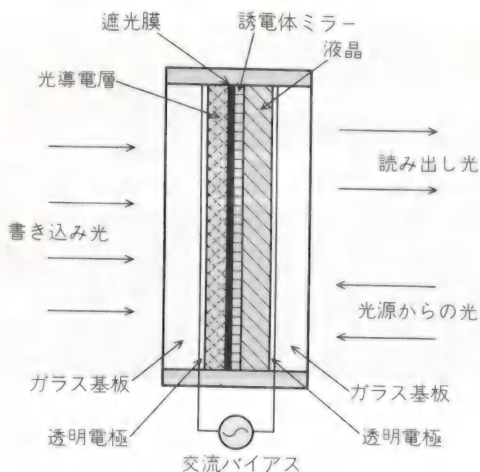
CRT方式は長年の実績から成熟度が高く，各種入力信号への追従特性も良好です。その反面，高輝度化するほどビーム径が太くなって，解像度が低下します。同時に蛍光面へのビーム突入によって発熱し，輝度と解像度に制限を受け，これを両立させることは，極めて困難です。

LCD方式は小型・軽量・量産に適しやすい特徴がある反面，解像度を上げると開口率(有効画素部の面積比率)が低下して，輝度の減少と非開口部(無効画素部)での光・熱変換によって発熱が起きます。この結果，輝度と解像度に制限を受け，その両立が困難なのはCRT方式と同様です。

両方式ともに改良されてはいるものの「輝度を上げると解像度が低下し，解像度を上げると輝度が低下する」という二律背反の課題を原理的に持っています。この対応として，複数台のプロジェクタ

【表1】
プロジェクタ方式の概要

方式	基本構成	特徴
CRT	映像信号 / 投写レンズ / CRT / スクリーン	・成熟度・従来システム適合性ともに高い ・画素構成なく，各種信号形式に適合しやすい ・高輝度化にはビーム量増大し，解像度の低下と発熱をともなう
LCD	ランプ / 映像信号 / LCDパネル / 投写レンズ / スクリーン	・単一パネルで量産に適合しやすい ・画素単位で構成され，入力信号形式が限定される ・光の利用率が低い ・高解像度化には開口率が下がり，輝度低下と発熱がともなう

※「ILA」は日本ビクター(株)の商標です。

【図1】
ILA用デバイスの基本構成

を用いたスタックやマルチスクリーンで市場要求に答えているのが実情です。

こうした現状に対して，この度開発されたILAスーパープロジェクタは，従来方式の原理的な課題を，空間光増幅デバイスを用いた新しい方式で解決し，二律背反していた高解像度と高輝度を初めて両立させた第3のディスプレイと言えます。

ILA用デバイス

ILA用デバイスは主として解像度を決定する画像入力系と，輝度を決定する出力系を分離させ，解像度と輝度の相互干渉を排除する構造となっています。このデバイスはプロジェクタ用途に限定されることなく，空間光情報処理分野の応用も進んでいます。

(1) デバイスの基本構成

デバイスは**図1**に示すとおり，2枚のガラス基板に挟まれた数層の薄膜だけで構成され，画素分割の微細加工がありません。各薄膜層は光導電層・遮光膜・誘電体ミラー・液晶となっており，その外側を透明電極で挟み，両電極間に交流バイアスが印加されます。

光導電層は微弱な2次元画像書き込み光に反応し，光電変換によって，液晶両端間電圧が制御されます。光導電層は書き込み系の感度，解像度，応答速度の決定要因となります。

遮光膜は原理的な基本動作には不要ですが，より強大な出力光を必要とするとき，高感度な光導電層に読み出し光の影響が発生しないよう，遮断を目的にしています。

誘電体ミラーは誘電体薄膜の積層構造で，液晶を通過してきた光源からの光を，ほぼ100%反射させ，読み出し光として出射させます。これによって光利用率の向上とともに，読み出し系と書き込み系との光学的遮断が行われ，デバイスの大きな特徴となります。

液晶は優れた電気光学効果を持ち，特に複屈折と誘電異方性が顕著で，印加電圧と偏光特性（光振動の偏り方）との関係で，光学特性に変化を与え，変調を行います。

デバイスは微弱な書き込み光に応じて，光導電層で光電変換によるインピーダンス変化が発生し，液晶両端に掛る電圧が制御されます。このとき，光源からの光は液晶を通り，誘電体ミラーで全反射され，再び液晶を経る間に光学的変調を受けます。この動作は書き込み光量に応じて，読み出し光の変調を制御しているにもかかわらず，誘電体ミラーで完全遮断することで，書き込み光と読み出し光の相互干渉が排除され，各々を独立させて取り扱えます。

この特徴は画像のリアルタイム処理に最適で，画像変換器，画像増幅器として応用できます。

画像変換器は，「赤外光入力－可視光出力」の波長変換，「自然光入力－レーザー光出力」の「I（非干渉）－C（可干渉）」変換，さらに書き込み，読み出しに各々異なった2次元画像を与える空間光論理演算など，光情報処理分野への応用も考えられています。

画像増幅器の代表的応用例は，プロジェクタで，高輝度ランプを読み出し光に用いれば，光増幅度百万倍以上が得られ，高輝度投写が可能となります。具体例としては静止画専用のCG用途の発表例があります。これを動画対応の汎用プロジェクタとするには，次の問題があります。

(1)光導電層の応答が遅く，ビデオレート（フィールド周波数60Hz）に追従できない

(2)液晶コントラスト比が低く，フルカラー画像に対応できない

これらの改善を次に示します。

(2) デバイス特性の改善

光導電層の必要条件は，高速応答，高感度，高解像度です。

各種光導電材料のうち，条件を満たすものにa-*Si*（アモルファスシリコン）があげられます。応答速度16m

sec以下で，60Hzのビデオレートに十分追従し，素材自体の電荷拡散が少ないので高解像度となり，感度も良好です。光導電層にa-Siを用いることで，書き込み系の特性改善がはかられています。

コントラスト比は，黒のレベルをいかに下げるかで決定されます。一般に，液晶の変調状態では波長依存性(例えばR，G，B別で変調状態が異なる)が生じ，可視光全波長帯域を同一変調にできず，黒レベルが浮く結果となります。これに対し，垂直配向の液晶は印加電圧が零のとき，無変調が得られ，波長依存性の伴なわない真の黒が得やすくなります。今回液晶(ネマティック)に垂直配向を行い，デバイス単体のコントラスト比は，数千以上が得られます。実際の配向は2度のプリティルト角を与え，動作方向の安定性を得ています。

光導電層と液晶の改善によって動画対応フルカラープロジェクタが可能となります。

(3) デバイス動作と特徴

図2を用いて，画像増幅器としてのILA用デバイスの動作を説明します。

光源からの光は偏りのない自然光で，PBS(偏向ビームスプリッタ)により，P波(図紙面に平行な波)とS波(図紙面に垂直な波)の直線偏向に分離されます。P波はPBSを直進，S波のみPBS界面で反射し，液晶に到達します。

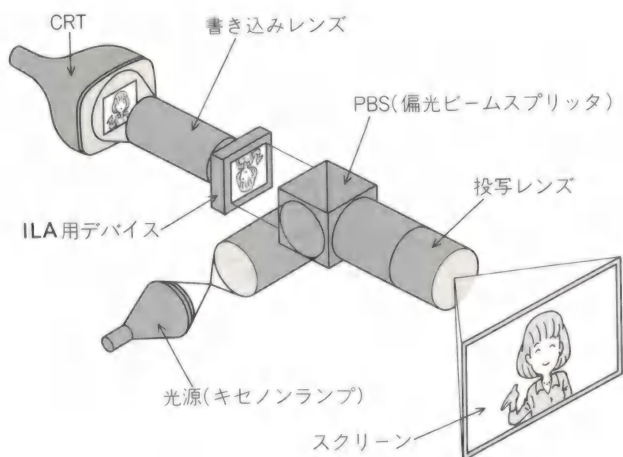
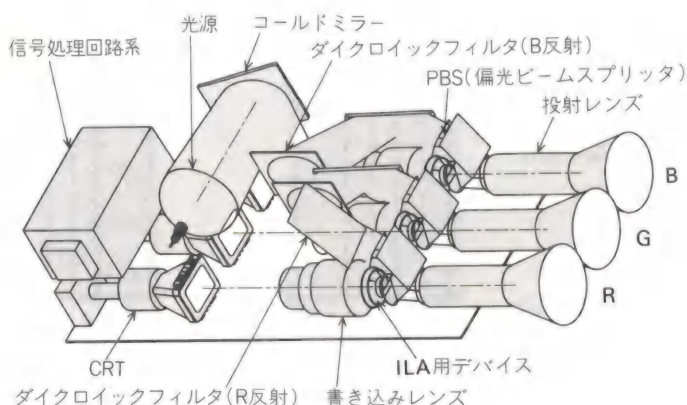
図3は垂直配向液晶分子の長軸方向と，光の変調特性との関係を示すもので，(a)は書き込み光無し，(b)は書き込み光有りの状態です。(a)では光導電層インピーダンスが高く，液晶両端電圧はしきい値以下となって，液晶分子長軸と通過光軸が合致し，複屈折効果を示しません。したがってS波で入射した光は，液晶・ミラー反射・液晶の経路を通った後もS波のまま反射して出射され，図2の入射時と同様経路で光源に戻るため，検出光側へは光が到達しません。この垂直配向による無変調状態は前記のとおり，波長依存性を持たず，高コントラスト比が得られます。

【図2】デバイス動作説明図

【図3】垂直配向液晶の動作

一方，光導電層が低インピーダンスとなる(b)では，液晶両端に数Vの電圧が印加されます。液晶分子は電界と直交する方向に傾き，通過光軸と液晶分子の長軸とが交り，複屈折性が現われます。S波の入射光は液晶を進む間に，偏光形態に変調を受け，図のとおり，楕円偏向や円偏光に変換された後，出射光となります。偏光形態は書き込み光量に応じて制御され，最大変調のとき，入射全光束はP波に変換され，PBSを通過して検出光となります。

以上のように光増幅器に用いたILAデバイスの原理的な特徴は次のとおりです。

(1)誘電体ミラーでほぼ100%反射し，高い光利用率と入出力間の光学的干渉を排除する
(2)熱吸収が極めて少なく，100万倍以上の光増幅率と高輝度出力が得られる
(3)垂直配向液晶の採用により，デバイス単体コントラスト比は，数千以上が得られる
(4)画素分割がなく，各薄膜の光拡散が少ないので，100 lp/mm以上の高解像度が得られる

【図4】プロジェクタの基本構成

【図5】ILA-M315Gの構成概要

(5)小型軽量，低電圧，低消費電力，高速応答など，アプリケーションが容易である

ILAスーパープロジェクタ

ILAスーパープロジェクタは，前記特徴を持つ空間光増幅デバイスを，初めて動画に対応するプロジェクタに応用し，従来二律背反していた，高輝度と高解像度を両立させ，2,000ルーメンの光出力と，1,000TV本の解像度を実現しています。

以下このILAプロジェクタの構成や性能について説明します。

(1) プロジェクタの基本構成

図4はプロジェクタの基本構成の一部でCRT書き込みとX_e(キセノン)ランプ光源を用いています。

書き込み光に要求される性能は，高解像度，光導電層に適合する発光波長，バリアブルスキャン対応および高安定度などです。CRTは成熟度が高く，安定度が良好で，画素分割がなく偏向回路のバリアブルスキャン化が容易で，蛍光体での発光波長の選定が可能などの特長を持っています。またCRTを低輝度で使用すれば，ビーム径が小さくなり，高い解像度が得られます。こうした適合性から，CRTを書き込み光に使っています。

CRT画像の投影は，波長帯域の限定とデバイスが高感度であることより，色収差や開口比からの規制が少なくなり，一般のリレー光学系で必要な特性が得られます。

光源からの光はCRT画像により制御され，PBSを直進した光だけが投写レンズによって拡大投写されます。

図5に，この度開発されたILAスーパープロジェクタ「ILA-M315G」の構成概要を示します。

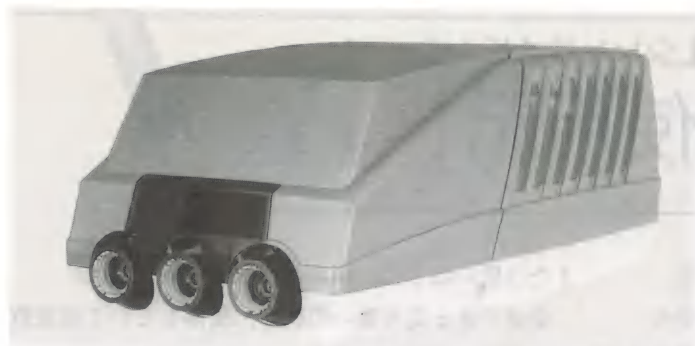
CRT，書き込みレンズ，デバイス，PBS及び投写レンズを，3原色R(赤)，G(緑)，B(青)用に，3組配置しています。

光源は高輝度化，演色性，安定度を考慮し，1,500Wのキセノン(X_e)ランプを採用しています。光源からの光は，集光ミラー，コンデンサレンズによって光束を整えた後，コールドミラーを介してダイクロイックフィルタによりR，G，B各々に分解されPBSに入射します。PBSは機械的変形や，熱的変形によって，特性が変化しないよう，多層薄膜基板を液体に浸した構造になっています。

投写レンズはR，G，Bが各々独立しているので，色収差の影響が少なく，解像度1,000TV本に対し十分な性能を持っています。

(2) ILA-M315Gの性能

ILA-M315Gスーパープロジェクタは，米国ヒューズ・エアクラフト社と日本ビクター㈱とで設立した合併会社HJT(ヒューズ・JVC・テクノロジー)社で開発・生産され，7月より日本国内販売を開始しています。その外観写真を**写真1**に示します。

ILA-M315Gは各種のテレビ方

〈写真1〉スーパープロジェクタILA-315G本体(左)とスクリーン(右)

式に対応できる業務用バリアブルスキャンプロジェクタ市場を対象にしたもので，本機の最大の特徴は，高輝度・高解像度を両立させたことです。従来のプロジェクタに対し，5倍以上にも達する2,000ルーメンの強力な光出力時にも，1,000TV本以上の高解像度が得られ，応答速度も16m sec以下で，ハイビジョン信号の投写にも最適なシステムです。

コントラスト比は250：1以上が得られ，ほぼ8ビットの最小分解能に相当します。この値は人間の知覚特性より決定される許容値に合致し，高画質映像に必要な条件を満たしています。

そのほか，広範な周波数に対する水平同期(15～90kHz)，垂直同期(45～120Hz)のオートロックおよび全世界の放送方式(NTSC/SECAM/PAL)デコーダの内蔵により各種入力信号への対応が可能です。

またコンバージェンスデータ，アスペクト比，画面サイズなどの投写条件，色調・階調，リトレースタイム(ブランキング幅)などの画像条件を記憶させる最大30チャネルのメモリーによって，プロジェクタ最適条件を数秒で設定します。

【表2】
ILA-M315Gの主な仕様

画面サイズ	90～450インチ
光出力	2000ルーメン
水平解像度	1000TV本以上
コントラスト比	250：1以上
水平同期周波数	15～90kHz(オートロック)
垂直同期周波数	45～120Hz(オートロック)
映像帯域	100MHz
アスペクト比	4：3～16：9(バリアブル)
入力信号	R, G, B コンポジット(NTSC/PAL/SECAM) Sビデオ
チャネルメモリ	最大30チャネル
ユーザーI/F	RS-232C
外　形	H520×W700×D1360(mm)
電　力	2500W
重　量	163kg

これらのチャネル制御はRS-232Cポートを介して行われ，同時に全システムのプログラマブル制御を可能にしています。

ILA-M315Gは，**表2**に示す仕様のとおり，業務用高級プロジェクタに要求される性能・機能を十分に満たし，ユーザーの要求に応える最大450インチの高画質・大画面を提供します。

以上のように**ILA**スーパープロジェクタは，従来「高輝度と高解像とが二律背反する問題」に対し，空間光増幅器を初めて動画対応プロジェクタに用いて高輝度・高解像度を両立させています。

ILA用デバイスの持つ本質的特徴は，高性能に加え，小型・軽量・量産適合性の良好さも見逃せない一面です。デバイス周辺の小型化開発も含め，近い将来各家庭で迫力ある大画面が楽しめるよう，民生用プロジェクタの開発も予定しています。

〈参考文献〉

① W.P.Bleha, "PROGRESS IN LIQUID CRYSTAL LIGHT VALVES" LASER FOCUS/ELECTRO-OPTICS (1983/10)
②若月，W.P.Bleha"ビデオ対応-SLM型液晶ライトバルブ プロジェクタ" ディスプレイ アンド イメージング1992，Vol.1 No.1
③中垣，W.P.Bleha "ILA™ビデオプロジェクタ"TV学会技術報告 Vol.17.No.10(1993/1)

(日本ビクター㈱プロジェクションシステム事業部技術部)

MW/SW/FMステレオ PLLシンセサイザ レシーバ

ICF-SW33の使用記

小林 良夫

◎ソニー

結論を先に言っておきます。実用的な放送受信，つまり，自分のいる地域を対象に放送している局の番組を手がるに楽しむうえで，使いやすく，実用上，十分な性能を備えている受信機です(**写真1**)。

あらかじめセットした17局をワンタッチで受信できる機能があります。これを最大に活用すれば，標準価格28,000円(税別)は安いでしょう。

単3型乾電池3本，またはAC電源(別売)などを使用，外形寸法は約16×9×3cm，重量は約400gと〝軽・小〟ですから，世界のどこへでも，ちょっと持って行く目的にも適しています。

ただ，受信できる放送の周波数は，地域によって違うため，購入時には何もセットしてありません。はじめに，自分でセットすることが必要です。

〈写真1〉ICF-SW33の外観

短波の放送は，季節によって周波数の変更があります。1つの場所で使っていても，周波数を1年に数回はセットしなおさなくてはなりません。

そのとき，取扱説明書は必需品――それというのも，この受信機はすべてマイコン制御で，1つのボタンが複数の動作をするなど，巧妙な設計です。すべてのボタンの，あらゆる動作を頭に入れるのは容易なことではありません。それより，はじめから取扱説明書を見てしまうほうが能率的です。

3種類の選局方法

この受信機は3つの方法で周波数の選択(つまり選局)ができるようになっています。

短波の受信を中心として，順に説明します。

①マニュアル選局：もっとも基本的な選局方式です。ダイヤル式受信機で，ダイヤルを動かすのと同じように，一定の間隔ごとに，周波数の高いほう，または低いほうに，順に受信していきます。

受信できる周波数の範囲と，受信していく周波数の間隔は，**表1**のようになっています。

短波の放送局は，大部分5kHz間隔で放送しており，kHzを単位とする周波数の数字は5の倍数になっています。しかし，半端な周波数の局もあるため，この受信機は1kHzごとに選べる設計です。

②スキャン選局：一定の強さ以上で受信できる放送を次々に拾っていく選局方法で，短波では，メータバンドごとにスキャンします。

いま受信している周波数から常に高いほうへ進み，放送が受かれば(厳密には，放送に限らず，電波が来ていれば…，ですが)，それを約2.5秒受信した後，また周波数の高いほうへ進んでいきます。

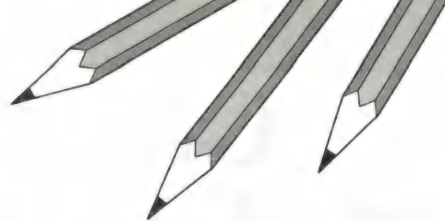
途中でスキャンを止めずに，そのバンドの上の端までいってしまうと，ふたたび下の端に戻り，また高いほうへ，バンドの中のスキャンが続きます。

③プリセット選局：あらかじめセットしてある周波数に相当するボタンを押すだけの，文字どおり，まさに〝ワンタッチ〟で受信する方式です。

最大で17局までセットできます。そのうち2局はスタンバイ操作(後述)のボタンに入れるもので，表の範囲の周波数の1つを自由に選べます。

残りの15局は，中波，短波，FM，それぞれ5局の配分です。この配分は変更できません。

短波の場合，例えば31mバンドの5局をセットしても，別々のバンドからでも，自由です。

いずれにしても，いきなりプリセット選局をするわけにはいきません。まず，①のマニュアル選局か②のスキャン選局で選んだ周波数をプリセットしてはじめて，プリセット選局の手順になります。

“お目ざめ”と“おやすみ”

この受信機には，スタンバイ(Standby。待機する意味)操作とスリープ(Sleep。こちらは，説明するまでもないでしょう)機能が付いています。

スタンバイ操作：a，b，2つのボタンがあって，時刻も，周波数も，各別にセットできます。

セットした時刻の10秒前から警告音が出て，10秒経つと，セットした放送の受信状態になります。

a，b，どちらかのボタンにお目覚めの時刻と，朝の受信に適した放送の周波数をセットしておけば目ざまし時計の役目に利用できるわけです。

スリープ機能：放送を受信しながらスリープ・ボタンを押すと，約60分でスイッチが切れます。

さらに時間を延ばしたいときは，この60分の間にもう1度スリープ・ボタンを押すと，そのときから約60分経過するまで，続いて受信ができます。

バンド	メータバンド	周波数範囲	周波数間隔
MW		531～1710 kHz	＊9kHz
SW	75m	3700～4200 kHz	1kHz（マニュアル選局）
	60m	4650～5150 kHz	
	49m	5800～6300 kHz	
	41m	6950～7450 kHz	
	31m	9375～10000kHz	
	25m	11525～12150kHz	5kHz（スキャン選局）
	21m	13375～14000kHz	
	19m	14975～15560kHz	
	16m	17475～18100kHz	
	13m	21320～21950kHz	
	11m	25475～26100kHz	
FM		76～108 MHz	0.05MHz

＊：10kHzステップに切り替えができる。

【表1】周波数範囲と周波数間隔

時差もわかる時計付き

この受信機には，正確な時計が付いており，自分のいる地域の時刻のほかに，世界の5地点の時刻がワンタッチで表示できます。

ただし，世界時との時差が1時間の整数倍でない地域に対する考慮は，してないようです。

例えば，JST(日本時間)は[世界時＋9時間]ですが，IST(インド時間。スリランカでも同じ)は[世界時＋5時間30分]となっています。

世界の時刻を表示するために，時差の数字(1時間単位)，および，東京，ロンドン，ニューヨークなど38の都市の名がプリセットしてあります。

ここでは，地名より，むしろ“UTC”(協定世界時)，“JST”(日本標準時)，“EST”(米国東部標準時)など，地域ごとの一般的な時刻の呼び名を入れておくのが本当でしょう。こうしてあれば，数百，あるいは，それ以上の都市の名をプリセットしてあるのと，結果的に，同じになったはずです。

ついでに言えば，短波のマニュアル選局が1kHz間隔になっているのも，いささか疑問です。

このため，選局の時間が長くなり，例えば11mバンドの下端の25,475kHzから上端の26,100kHzに行きつくまで，どうしても約1分かかります。

しかも，マイコンの動作時間が影響しているようで，表示窓に周波数の数字が出た瞬間に受信状態になるわけではなく，実際に，放送の内容が音として出て来るまでに，1秒弱の遅れがあります。

したがって，1kHz間隔でどんどん周波数を上げ，あるいは下げている間，放送は受信できません。

何か放送があると思ったら，少なくても2～3秒は選局を止めて，受信してみる必要があります。

主要な短波の放送局は，すべて5kHzごとに出ていますから，むしろ，5kHz間隔で選局する方式に割り切ってしまいたいところです。こうしておけば選局のスピードも数倍に早くなったはずです。

付け加えれば，半端な周波数の放送が，実際にピッタリその周波数に出ているとは限りませんし，本当に半端な周波数に出ているときでも，それに近似の5kHzの倍数の周波数で受信して，実用上，ほとんど支障はないのですから…。

しかし，原則としてプリセット選局を使う限り，全体として，使いやすい受信機といえるでしょう。

C言語プログラムで学ぶ ビット・リダクション講座

大富 志太

6 IDCT(逆離散コサイン変換)

先月号では，いろいろな画像信号をDCT係数に変換して，どの程度ビットリダクションができるか，グラフィック表示して確かめてみました。もちろん，MPEG規格を使った画像圧縮の応用機器が，DCT係数をそのまま伝送したり，記録したりしているわけではありません。せっかくDCTのプログラムを作っても，最後までビット圧縮の効果がわからないのでは面白くありませんので，あくまでも目安として，DCTの効果をつかんでいただくための試みです。

したがって，圧縮度の正確さはありませんが，DCTにとってどのような種類の画像がやさしく，どのような画像が難しいかがおわかりいただけたのではないかと思います。

今月号では，DCT係数を逆離散コサイン変換をして，元の画像に直すプログラムと，復元した画像が元の画像に比べてどの程度の誤差を含んでいるかを調べるプログラムを作ってみましょう。

逆DCTの計算式

DCT係数に変換された画像データは，受信側に伝送され，元の画像に復元されることになります。伝送のときには，DCT係数を直接伝送しないで，後でお話するような，係数をゼロ・ランレングス化するなどの操作が入りますが，最終的には逆DCT変換して元の画像を再現することになります。そして，MPEGで採用している伝送のための符号化は，ハフマン符号化といわれるもので，符号化によるひずみはありません。

今回は，ひずみを発生しない部分はスキップして，直接元の画像に戻す逆DCTのプログラムにチャレンジしてみましょう。

MPEGの逆DCT変換式は，1式に示すものです。

$$f(i,\ j)=(1/4)\sum_{u=0}^{7}\sum_{v=0}^{7}c(u)c(v)F(u,v)\cos(2i+1)u\pi/16\cos(2j+1)v\pi/16 \quad \cdots\cdots 1式$$

左のf (i, j)は逆変換された画像を，右辺のF (u, v)はDCT係数を示します。iは，復元される画像の垂直方向の画素位置を，jは水平方向の画素位置を表し，uとvが，それぞれDCT係数の垂直と水平の位置を表していることは，6月号のDCTの式と同じです。uとvは，両方とも0から7まで変化し，数が多くなるほど周波数が高くなります。また，cos(2i+1)uπとcos(2j+1)vπは，変換される画像が持っている周波数成分を表し，その大きさはF (u, v)の値で決定されます。

c (u)とc (v)は，u=0とv=0で$1/\sqrt{2}$，uとvが1以上で1.0であることもDCTの式と同じです。

逆DCTのプログラム

この1式をC言語に直してみたプログラムが**リスト1**です。それでは，プログラムを細かくみていきましょう。c(u)とc(v)をcu[u]

【リスト1】
逆DCT変換式をC言語に直したプログラム

```
cu[0] = cv[0] = 1.0/sqrt(2.0);                    /* 補正係数の初期化 */
for(i=1; i<8; i++)
    cu[i] = cv[i] = 1.0;

for(i=0; i<8; i++){
    for(j=0; j<8; j++){                            /* 画像データの計算 */
        picture[i][j] = 0.0;
        for(u=0; u<8; u++){
            for(v=0; v<8; v++){
                c2 = cos((2.0*(float)i+1.0)*(float)u*PAI/16.0);
                c3 = cos((2.0*(float)j+1.0)*(float)v*PAI/16.0);
                picture[i][j] += cu[u]*cv[v]*(float)coeff[u][v]*c2*c3;
            }
        }
        picture[i][j] *= (1.0/4.0);
        if(picture[i][j]<0)                        /* 画像データの整数化 */
            picture[i][j] -= 0.499999;
        else
            picture[i][j] += 0.499999;
        pictureint[i][j] = (int)picture[i][j];
    }
}
```

とcv［v］したのは，この前の説明のとおりです。初めはcu［0］とcv［0］を$1/\sqrt{2}$に初期化しています。次に，for文を使ってcu［1］～cu［7］とcv［1］～cv［7］を1.0に初期化します。

次に，逆DCT変換の中心的なプログラムに取りかかりましょう。左辺の小文字のf(i, j)の中では，iとjが変化する変数で，右辺のF(u, v)は，uとvが変数になります。したがって，for文を使ってループを4つ作ります。左辺の変数をもつループを先に，右辺の変数をもつループを後におきます。全体では**リスト2**のようになります。

変換が終わった64の画像データを蓄えるために，float型のpicture［8］［8］の配列と，整数に変換した画像データを入れるため，pictureint［8］［8］の配列を用意します。

式には現れませんが，picture［i］［j］配列を，Σの外側でゼロにする必要があります。これが，picture［i］［j］=0.0；の文になります。

Σの中側は，1つの文で表しても構いませんが，数式のどこにどの部分が対応しているのかわかりやすくするため，式を3つに分けることにします。cos関数が2つありますので，最初，float型のc2とc3と名づけた変数に，おのおののcos関数の値を計算し代入します。この部分は，

c2＝cos((2.0＊(float)i＋1)＊(float)u＊PAI/16.0);
c3＝cos((2.0＊(float)j＋1)＊(float)v＊PAI/16.0);

となります。

(float)は，整数型の変数の値を，float型に変換するためのものです。次に，求められたc2とc3を使って，coeff［u］［v］配列に入れられたDCT係数を画像データに変換します。

picture［i］［j］＋＝cu［u］＊cv［v］＊(float)coeff［u］［v］＊c2＊c3;

cu［u］とcv［v］は，uとvにしたがって先ほど初期化したものを使います。ゼロに初期化された，picture［i］［j］配列には，uとvを変化させて計算した値が，＋＝によって次々に加算されます。uとvのループが終わったら，Σの外にある4分の1を実行するための掛け算をします。

これが，

picture［i］［j］＊＝(1.0/4.0);

です。

【リスト2】
for文を使ってループを4つ作る

```
for(i=0; i<8; i++){
        for(j=0; j<8; j++){
                Σの外に出ている部分の処理文
                for(u=0; u<8; u++){
                        for(v=0; v<8; v++){
                                Σの内側の処理文
                        }
                }
        }
}
```

picture[i][j]に計算されたfloat型の画像データを，小数点以下第1位で四捨五入して，整数配列のpictureint [i] [j] に代入します。これは，if文を使って正の場合と負の場合に分けて，正の場合は，0.499999を加算，負の場合は減算したうえで，型変換をします。

```
if(picture [i] [j] < 0)
    picture [i] [j]-=0.499999;
else
    picture [i] [j]+=0.499999;
pictureint[i][j]=(int)picture[i][j];
```

これで，逆DCTの計算が終わりました。

ひずみの計算式

画像データをDCTで変換して得た係数を，逆DCTで再び画像データに戻したとき，元の画像データとまったく同じ値になったときには，変換によるひずみは発生しなかったといえます。元のデータと差がある場合には，DCT変換と逆変換で誤差が発生したことになります。この誤差を信号対雑音比，SNR(Signal to Noise Ratio)として表すのが普通になっています。

読者の皆さんは，音声信号のひずみ率を規定した数式をご覧になられたことがあると思いますが，SNRもひずみ率を求める方法と同じです。

ひずみ率の場合には，高調波の自乗和の平方根を基本波で割って求めました。SNRも，64の画素の誤差の自乗和の平方根を，元の画素レベルの総和で割った値と規定します。元の画像をf(i, j)復元画像をf'(i, j)とすると，SNRは，

$$SNR=\sqrt{\sum_{i=0}^{7}\sum_{j=0}^{7}[f(i,j)-f'(i,j)]^2} / \sum_{i=0}^{7}\sum_{j=0}^{7}f(i,j) \cdots 2式$$

で示されます。

SNRのプログラム

計算式をプログラムに変換してみましょう。元の画像データの配列をpicture[i] [j]，復元された画像データをpictureint[i][j]，SNRをsnrとして表すことにします。プログラムをわかりやすくするため，分母をdenominator，分子をnumeraterという変数として，まとめることにします。

例によってfor文でループを作りましょう(**リスト3**)。

これでSNRを求めることができます。forループのプログラム文を実行する前に，denominatorとnumeraterの2つの変数を初期化しておく必要があります。もちろん，初期化の文を単独に作ってもいいのですが，for文で，カウンタとなるiをゼロに初期化する際，2つの変数も初期値を入れることができます。初期化したい変数を「,」で区切って幾つでも初期化できます。あまりきれいではないと思われる方は，普通文を使って初期化してください。

SNRを求めるとき，分母がゼロになる場合にはエラーになるため計算式を実行してはなりません。if文を使って分母がゼロでないときには割り算を実施し，ゼロの場合には，SNRに0.0を代入します。

表示まで含んだプログラム

基本的なプログラムは，これまで説明したとおりです。DCT変換と逆DCTをして求められたひずみと復元画像は，おのおの用意された配列の中に格納されていますが，そのままでは中身を見ることができません。プログラムの量は少し大きくなりますが，中身を表示するプログラムを追加してみました。

全プログラムを**リスト4**に示します。

先月号で紹介した，普通の画像，平坦なグレー画像など，8種類の画像データをメニュー画面から選択し，DCTと逆DCTを実行します。8×8のブロックの中で，どのようにひずみが発生しているか

```
for(i=0,denominator=0.0,numerater=0.0; i<8; i++){
    for(j=0; j<8; j++){
       denominator+=pow((float)(picture[i][j]-pictureint[i][j]),2.0);
       numerater+=picture[i][j];
    }
}
if((int)numerater != 0)
   snr=denominator/numerater;
else
   snr=0.0;
```

【リスト3】
for文でループを作る

の様子と，SNRを比率とデシベルで表示します。

また，変換前と変換後の画像データを，ひずみによる変化がわかるように3次元表示します。細かなプログラムの説明は省略しますが，このとおり打ち込んでコンパイルすれば動作しますので，ご自分で確かめて見られることをお勧めします。

〈写真 1〉
メニュー画面

◆

プログラムの実行結果

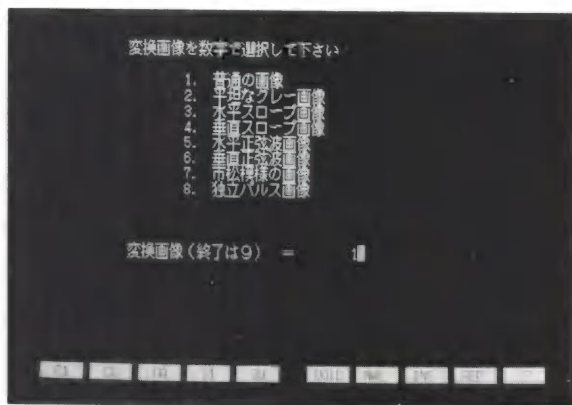
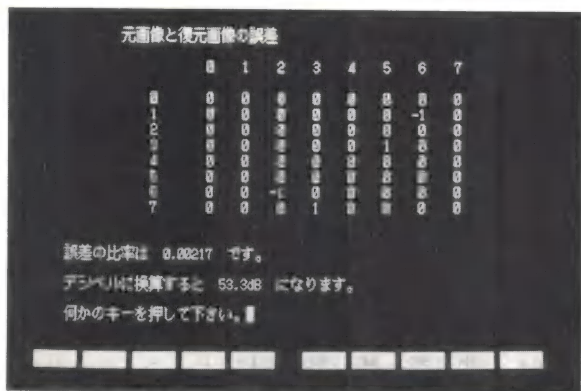
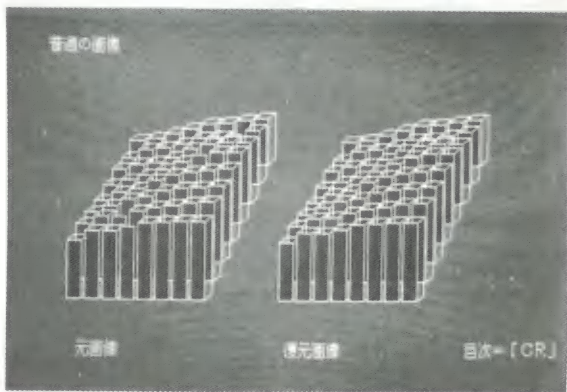
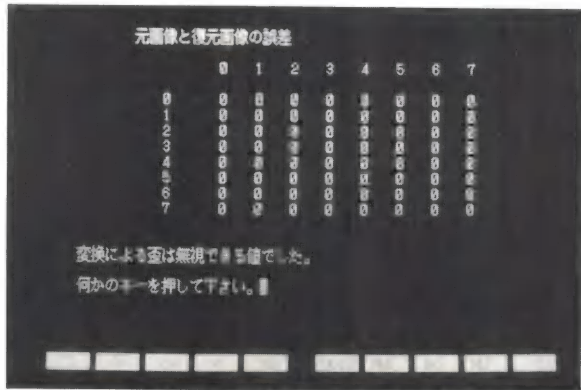
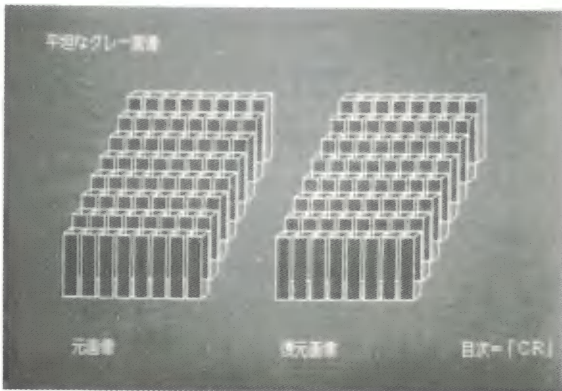
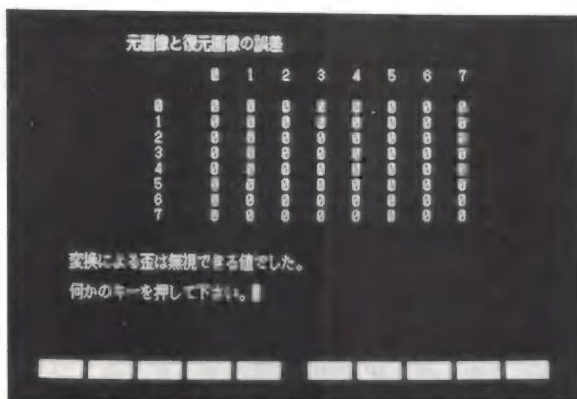
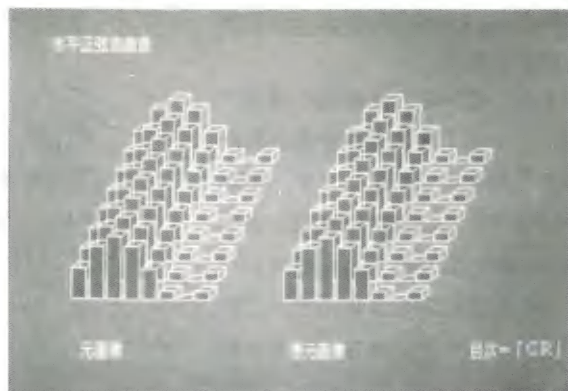
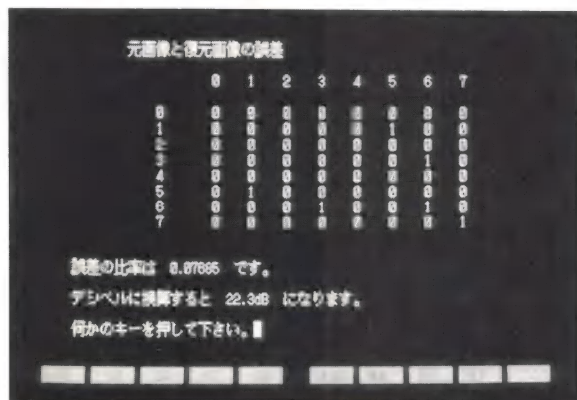
プログラムを走らせると，**写真1**のようなメニュー画面が現れます。変換する画面を数字で選択すると，DCTと逆DCTを実行し，結果から元画像と復元画像の間の誤差をマトリックスで表示し，SNRを比率とデシベルで表示します。何かキーをたたくと，次の画面に進みます。次の画面は，元画像と復元画像を3次元グラフィック表示します。何かのキーをたたけば，最初のメニュー画面に戻ります。

それでは，表示結果をご紹介しましょう。

写真2は，日ごろのテレビ放送によく現れるような「普通の画像」のひずみです。ひずみの率は，0.00217で，53.3dBですから，劣化がほとんど問題にならない画像といえます。**写真3**は元の画像と復元画像を並べて表示したものです。復元画像のトップに濃く表示されている部分がひずみです。

写真4と**写真5**は，一面「グレーの画像」のひずみと，画像の3次元表示です。極めて簡単な画像で，当然，ひずみはまったく発生しません。**写真6**と**写真7**は，「水

〈写真 2〉「普通の画像」のひずみ

〈写真 3〉 元の画像と復元画像の表示

〈写真 4〉「グレーの画像」のひずみ

〈写真 5〉 グレーの画像

〈写真6〉「水平の正弦波」のひずみ

〈写真7〉 水平の正弦波

〈写真8〉
SNRを比率と
デシベルで表示

平の正弦波」のデータですが，変換の基準となる余弦関数と親戚関係にあるため，誤差は発生していません。

8種の画像データの中，最もひずみの発生しやすかったのは，単一のパルスです。**写真8**に出ているように，ひずみは0.07685，22.3dBになっています。数字としては，大きなひずみが出ているようですが，ひずみの定義式でわかるように，単一パルスであるため分母の値が小さくなったためで，ひずみそのものの発生は，普通の画像と変わっていません。

これまでで，MPEGの画像圧縮技術の中心的なDCTの変換と逆DCT変換を見てきました。次回は，ハフマン符号化に話を進める予定でいます。

```
/* This program is for transforming a block of pictrue data defined
   by 8*8 pixcels into DCT coefficients by using the MPEG DCT formula.
   DCT coefficients are again transformed to picture data by the MPEG
   inverse formula and their difference values from the original ones
   are calculated as a Signal to Noise Ratio.
*/


#include     <stdio.h>
#include     <math.h>
#include     <conio.h>
#include     <alloc.h>
#include     <graphics.h>

#define PAI 3.14159

main(){

char         msg[8][20]={ "普通の画像",
                          "平坦なグレー画像",
                          "水平スロープ画像",
                          "垂直スロープ画像",
```

【リスト4】表示を含んだプログラム

```
                             "水平正弦波画像",
                             "垂直正弦波画像",
                             "市松模様の画像",
                             "独立パルス画像"};
char          inbuf[10];

int  picture[8][8][8]={ 47, 50, 61, 71, 81, 83, 88, 93,
                        51, 69, 72, 81, 92, 95,100,108,
                        68, 78, 82, 92,111,120,124,125,
                        75, 86, 96,108,125,139,144,145,
                        82,100,109,126,139,144,149,153,
                        98,112,127,132,144,151,153,156,
                       115,130,131,143,150,155,160,163,
                       121,140,141,150,159,161,162,163,

                       128,128,128,128,128,128,128,128,
                       128,128,128,128,128,128,128,128,
                       128,128,128,128,128,128,128,128,
                       128,128,128,128,128,128,128,128,
                       128,128,128,128,128,128,128,128,
                       128,128,128,128,128,128,128,128,
                       128,128,128,128,128,128,128,128,
                       128,128,128,128,128,128,128,128,

                        48, 68, 88,108,128,148,168,188,
                        48, 68, 88,108,128,148,168,188,
                        48, 68, 88,108,128,148,168,188,
                        48, 68, 88,108,128,148,168,188,
                        48, 68, 88,108,128,148,168,188,
                        48, 68, 88,108,128,148,168,188,
                        48, 68, 88,108,128,148,168,188,
                        48, 68, 88,108,128,148,168,188,

                       188,188,188,188,188,188,188,188,
                       168,168,168,168,168,168,168,168,
                       148,148,148,148,148,148,148,148,
                       128,128,128,128,128,128,128,128,
                       108,108,108,108,108,108,108,108,
                        88, 88, 88, 88, 88, 88, 88, 88,
                        68, 68, 68, 68, 68, 68, 68, 68,
                        48, 48, 48, 48, 48, 48, 48, 48,

                         64,109,128,109, 64, 18,  0, 18,
                         64,109,128,109, 64, 18,  0, 18,
                         64,109,128,109, 64, 18,  0, 18,
                         64,109,128,109, 64, 18,  0, 18,
                         64,109,128,109, 64, 18,  0, 18,
                         64,109,128,109, 64, 18,  0, 18,
                         64,109,128,109, 64, 18,  0, 18,
                         64,109,128,109, 64, 18,  0, 18,

                         64, 64, 64, 64, 64, 64, 64, 64,
                        109,109,109,109,109,109,109,109,
                        128,128,128,128,128,128,128,128,
                        109,109,109,109,109,109,109,109,
                         64, 64, 64, 64, 64, 64, 64, 64,
                         18, 18, 18, 18, 18, 18, 18, 18,
                          0,  0,  0,  0,  0,  0,  0,  0,
                         18, 18, 18, 18, 18, 18, 18, 18,

                        128,128,  0,  0,128,128,  0,  0,
                        128,128,  0,  0,128,128,  0,  0,
                          0,  0,128,128,  0,  0,128,128,
                          0,  0,128,128,  0,  0,128,128,
                        128,128,  0,  0,128,128,  0,  0,
                        128,128,  0,  0,128,128,  0,  0,
                          0,  0,128,128,  0,  0,128,128,
                          0,  0,128,128,  0,  0,128,128,

                          0,  0,  0,  0,  0,  0,  0,  0,
                          0,  0,  0,  0,  0,  0,  0,  0,
                          0,  0,  0,  0,  0,  0,  0,  0,
                          0,  0,  0,  0,  0,  0,  0,  0,
                          0,  0,  0,  0,  0,  0,  0,  0,
                          0,  0,  0,  0,  0,255,  0,  0,
                          0,  0,  0,  0,  0,  0,  0,  0,
```

【リスト4】表示を含んだプログラム(つづき)

```
                              0,  0,  0,  0,  0,  0,  0,  0};

    int      dctint[8][8], ret_picture[8][8], error[8][8],
             n=99, u, v, i, j;
    float    fl_picture[8][8], dctcoff[8][8], cu[8], cv[8], c1, c2, c3,
             denominator, numerater, total_error, log_error;


    do{
        clrscr();
        printf("¥n¥n¥n¥t¥t変換画像を数字で選択して下さい¥n¥n");

        printf("¥t¥t¥t1.  普通の画像¥n");
        printf("¥t¥t¥t2.  平坦なグレー画像¥n");
        printf("¥t¥t¥t3.  水平スロープ画像¥n");
        printf("¥t¥t¥t4.  垂直スロープ画像¥n");
        printf("¥t¥t¥t5.  水平正弦波画像¥n");
        printf("¥t¥t¥t6.  垂直正弦波画像¥n");
        printf("¥t¥t¥t7.  市松模様の画像¥n");
        printf("¥t¥t¥t8.  独立パルス画像¥n¥n");

        printf("¥n¥n¥t¥t変換画像(終了は9)   =¥t");
        gets(inbuf);
        sscanf(inbuf, "%d", &n);
        n -= 1;
        if(n==8)
            exit(0);
       cu[0] = cv[0] = 1.0/sqrt(2.0);              /* 補正係数の初期化 */
       for(i=1; i<8; i++)
           cu[i] = cv[i] = 1.0;

       for(u=0; u<8; u++){                         /* DCT係数の計算 */
           for(v=0; v<8; v++){
               dctcoff[u][v] = 0.0;
               c1 = (1.0/4.0)*cu[u]*cv[v];
               for(i=0; i<8; i++){
                   for(j=0; j<8; j++){
                       c2 = cos((2.0*(float)i+1.0)*(float)u*PAI/16.0);
                       c3 = cos((2.0*(float)j+1.0)*(float)v*PAI/16.0);
                       dctcoff[u][v] += (float)picture[n][i][j]*c2*c3;
                   }
               }
               dctcoff[u][v] *= c1;
               if(dctcoff[u][v]<0)                 /* DCT係数の整数化 */
                   dctcoff[u][v] -= 0.499999;
               else
                   dctcoff[u][v] += 0.499999;
               dctint[u][v] = (int)dctcoff[u][v];
           }
       }

   /*-----------------------------------------------------------------*/

   /* This program is for transforming a block of DCT coefficients into
      picture data by using the MPEG IDCT formula */


       cu[0] = cv[0] = 1.0/sqrt(2.0);              /* 補正係数の初期化 */
       for(i=1; i<8; i++)
           cu[i] = cv[i] = 1.0;

       for(i=0; i<8; i++){                         /* 画像データの計算 */
           for(j=0; j<8; j++){
               fl_picture[i][j] = 0.0;
               for(u=0; u<8; u++){
                   for(v=0; v<8; v++){
                       c2 = cos((2.0*(float)i+1.0)*(float)u*PAI/16.0);
                       c3 = cos((2.0*(float)j+1.0)*(float)v*PAI/16.0);
                       fl_picture[i][j] += cu[u]*cv[v]*(float)dctint[u][v]*c2*c3;
                   }
               }
               fl_picture[i][j] *= (1.0/4.0);
               if(fl_picture[i][j]<0)                  /* 画像データの整数化 */
                   fl_picture[i][j] -= 0.499999;
```

【リスト4】表示を含んだプログラム(つづき)

```
                else
                    fl_picture[i][j] += 0.499999;
                ret_picture[i][j] = (int)fl_picture[i][j];
            }
        }

    /*------------------------------------------------------------*/

        for(i=0,denominator=0.0,numerater=0.0; i<8; i++){
            for(j=0; j<8; j++){
                denominator += (float)picture[n][i][j];
                error[i][j] = picture[n][i][j] - ret_picture[i][j];
                numerater += (float)(error[i][j]*error[i][j]);
            }
        }

       if(denominator == 0.0){
           total_error = 0.0;
           u = 1;
       }
       else{
           total_error = sqrt(numerater/(8.0*8.0))/(denominator/(8.0*8.0));
           if(total_error != 0.0){
               log_error = 20.0*log10(1.0/total_error);
               u = 0;
           }
           else {
               u = 2;
           }
       }

       clrscr();
       printf("¥n¥n¥n¥t¥t元画像と復元画像の誤差¥n¥n¥t¥t          ");
       for(i=0; i<8; i++)
           printf("%5d",i);
       printf("¥n¥n");
       for(i=0; i<8; i++){
           printf("¥t¥t%5d    ",i);
           for(j=0; j<8; j++)
               printf("%5d",error[i][j]);
           printf("¥n");
       }

       if( u == 1 ){
           printf("¥n¥n¥t変換画像の振幅はゼロでした。¥n¥n");
           continue;
       }

       if( u != 2){
           printf("¥n¥n¥t誤差の比率は  %5.5f  です。¥n¥n",total_error);
           printf("¥tデシベルに換算すると  %3.1fdB  になります。¥n¥n",(log_error+0.04
       }
       else
           printf("¥n¥n¥t変換による歪は無視できる値でした。¥n¥n");

       printf("¥t何かのキーを押して下さい。");
       getch();

       clrscr();                                /* 計算結果の表示 */
       graph3(picture, ret_picture, error, msg[n], n);

       } while( n<9 );
    return 0;
    }

    /*------------------------------------------------------------*/
    graph3( array3, array4, array5, msg, n )

    char    msg[20];
    int     *array3, *array4, *array5, n;

    {
        int i, j,
            left, top, right, bottom, depth = 8, topflag,
            graphdriver = DETECT, graphmode = PC98C16,
            start_x = 140, start_y = 160,
```

【リスト4】表示を含んだプログラム(つづき)

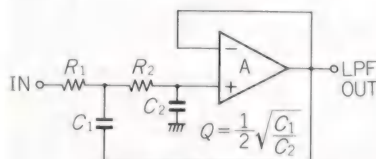
```
        step = 20, width = 16, slant = 10, s = 2;
    detectgraph( &graphdriver, &graphmode );
    initgraph( &graphdriver, &graphmode,"A:¥BOLANDC¥BGI" );
    setbkcolor( BLUE );
    cleardevice();
    moveto( 48, 32 ); outtext( msg );

    for( i=0; i<8; i++ ){
        for( j=0; j<8; j++ ){
            moveto( start_x + step*j, start_y + step*i );
            left = start_x + step*j - slant*i;
            right = left + width;
            bottom = start_y + step*i;
            top = bottom - *( array3 +n*64 + i*8 + j )/s;
            if( *( array5 + i*8 + j ) == 0 )
                topflag = 1;
            else
                topflag = 0;
            if( *( array3 +n*64 + i*8 + j ) >= 0 )
                setfillstyle( SOLID_FILL, GREEN );
            else
                setfillstyle( SOLID_FILL, RED );

            bar3d( left, top, right, bottom, depth, topflag );
        }
    }

    start_x = 380;
    for( i=0; i<8; i++ ){
        for( j=0; j<8; j++ ){
            moveto( start_x + step*j, start_y + step*i );
            left = start_x + step*j - slant*i;
            right = left + width;
            bottom = start_y + step*i;
            top = bottom - *( array4 + i*8 + j )/(s);
            if( *( array5 + i*8 +j ) == 0)
                topflag = 1;
            else
                topflag = 0;

            if( *( array4 + i*8 + j ) >= 0 )
                setfillstyle( SOLID_FILL, GREEN );
            else
                setfillstyle( SOLID_FILL, RED );

            bar3d( left, top, right, bottom, depth, topflag );

            if( *( array5 + i*8 +j ) != 0 ){
                bottom = top;
                top = bottom - *( array5 + i*8 + j )/(s);
                topflag = 1;

                if( *( array5 + i*8 + j ) >= 0 )
                    setfillstyle( CLOSE_DOT_FILL, WHITE );
                else
                    setfillstyle( CLOSE_DOT_FILL, YELLOW );

                bar3d( left, top, right, bottom, depth, topflag );
            }

        }
    }
        moveto( 80, 344 ); outtext( "元画像" );
        moveto( 300, 344 ); outtext( "  復元画像" );
        moveto( 520, 344 ); outtext( "目次=「CR」");
        getch();
        setbkcolor( BLACK );
        cleardevice();
        closegraph();
        return 0;
}

/*----------------------------------------------------------------*/
```

【リスト4】表示を含んだプログラム(つづき)

稲葉 保

同一回路定数で構成する実用的なアクティブ・フィルタ

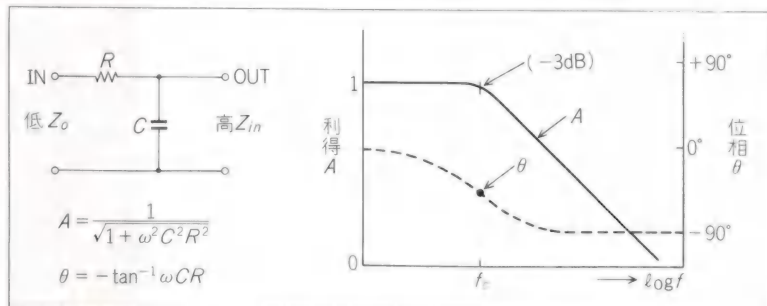
アクティブ・フィルタを製作しようとするとき，コンデンサの入手が面倒な場合があります。例えば，**図1**のようなローパス・フィルタ(以下LPFと略す)では，コンデンサC_1とC_2の比率を$\sqrt{2}C_1:C_2/\sqrt{2}$としなければなりません。

入手しやすい値としてE6～E12系列を採用するので，ちょうど2：1とするには1.5×10^{-n}と7.5×10^{-n}しかありませんが，さらに一般性を考慮するとE6系列で設計したいが，2：1の比率は存在しません。

そこで，コンデンサCや抵抗Rの値を等しくできるフィルタ回路について解説，実験してみます。

CR回路網の特性

低周波回路で使われるフィルタのほとんどがアクティブ・フィルタで，抵抗，コンデンサにOPアンプなどを組み合わせて実現します。

はじめにCR回路のみで構成するフィルタの性質について説明します。**図2**は最も簡単なCR型ローパス・フィルタで，遮断周波数f_Cより高い周波数を減衰します。

遮断周波数f_Cとは通過域を基準に，これより振幅が$1/\sqrt{2}$に低下する周波数を言い，

$$A=1/\sqrt{1+\omega^2C^2R^2}$$

より，分母が$\sqrt{2}$となるためには，

$\omega^2C^2R^2=1 \quad \omega=2\pi f,$

$$(2\pi f)^2=1/C^2\cdot R^2$$

$$\therefore f=1/2\pi CR \Rightarrow f_C=1/2\pi CR$$

となじみの多い式となります。

振幅(利得A)が$1/\sqrt{2}$になるということは，デシベル表記すると，

$$20\log_{10}(1/\sqrt{2})=-3\text{dB}$$

となります(f_Cのことを－3dBポイントとも言う)。

CR型フィルタでは減衰特性が悪く(6dB/oct)，入力周波数が2倍(オクターブ)になると出力振幅が1/2にしか減衰しません。

フィルタ回路は，振幅の減衰と同時に入出力間の位相関係も変化します。LPFの場合は，遮断周波数f_Cより極めて低い入力周波数では位相遅れを生じませんが，図2のようにf_Cに近づくにしたがい遅れ，$f=f_C$にて$-45°$，f_C以上ではやがて$-90°$に達します。

任意の入力周波数fでの位相角θは，$\theta=-\tan^{-1}\omega CR$より計算でき，$\omega=CR$では，

$$\theta=-\tan^{-1}\cdot 1=-45°$$

になります。

CR型のフィルタの欠点には，先ほど述べた減衰特性の悪さのほかに周辺回路インピーダンスに影響される点です。つまり，フィルタを駆動するインピーダンスZ_oは抵

【図1】標準的な12dB/octLPF

【図2】CR回路網の特性

抗Rに対して十分小さく，受側の入力インピーダンスZ_{in}は抵抗Rに対して十分大きくないと，遮断周波数f_Cが変化してしまいます。

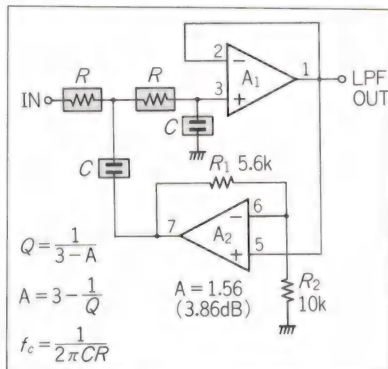
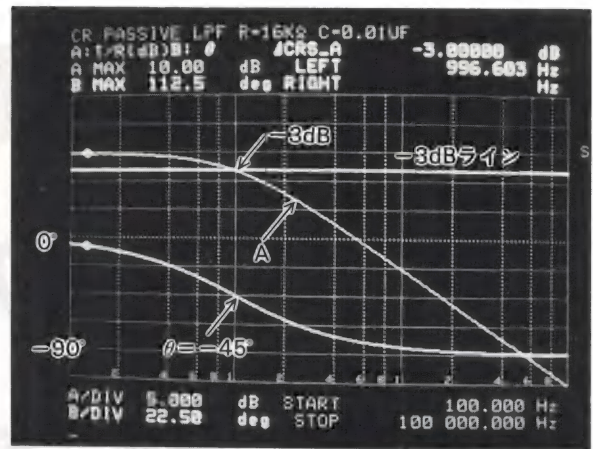
写真1は$Z_0=50\Omega$，$Z_{in}=1\,M\Omega$，$R=16k\Omega$，$C=0.01\mu F$（$\therefore f_C=1$ kHz）での振幅，位相特性です。使用したCRの定数は計算値どおりではないので，$f_C=996.6$Hzとなっていますが，これならほぼ計算どおりと言えます。

一方，位相特性は1目盛が22.5°ですから，f_Cにて$-45°$です。

〈写真1〉
CR型フィルタの
振幅A，位相θ特性

最も基本的なアクティブ・フィルタ

図1は抵抗2本（$R_1=R_2$）とコンデンサ2本（$C_1 \fallingdotseq C_2$）に，OPアンプ・バッファを付加した12dB/oct LPFで，CR型より減衰特性を良くするため回路のQ（f_C付近のシャープさを示すファクタ）を大きくしています。

12dB/octのフィルタでは，Qを$1/\sqrt{2}=0.707$に設定すると最平坦な特性（これをバタワース応答と言う）が得られますが，どうやってQを上げるかと言うと，コンデンサC_1をOPアンプ出力で駆動（正帰還）し，C_1とC_2にアンバランスな容量を設定してやります。

【図3】同一定数で構成した12dB/octLPF

このときのQは
$Q=1/2\cdot(\sqrt{C_1/C_2})$で，
両辺を2乗して整理するとC_1の比率が求められ，

$C_1=4Q^2\cdot C_2$，ここで

$Q=1/\sqrt{2}=2\cdot C_2$

となり，C_1はC_2の2倍の容量としなければなりません。

同一定数にするには

同一定数とするには，$C_1 \geqq C_2$として正帰還量を設定するのと同等な方法は，**図3**のように帰還ループに利得を持つアンプA_2を入れます。利得Aが1より大きくなると回路のQは増大し，A＝3に近づくと不安定になり，やがて発振します。

Qの設定は利得Aで行うため正確な利得設定を要求され，Qが決まれば，$A=3-\frac{1}{Q}$より算出でき，12dB/octでバタワース応答なら$Q=0.707$ですから，

$A=3-1.41=1.59$倍

$R_1=R_2(A-1)=5.9k\Omega$

となり，$R_1=5.6k\Omega$とします。

これで抵抗Rを等しく（本来，等しい），コンデンサCも等しくすることができます。設計の手順としては，遮断周波数f_Cが決まれば都合の良いコンデンサ（CのリアクタンスX_Cが10k～100kΩになる値で$X_C=1/2\pi f_C\cdot C$）を選び，

$R=1/2\pi f_C\cdot C$

で計算し，$f_C=3$kHz，$C=2{,}200$pFでは，$R \fallingdotseq 24k\Omega$となります。

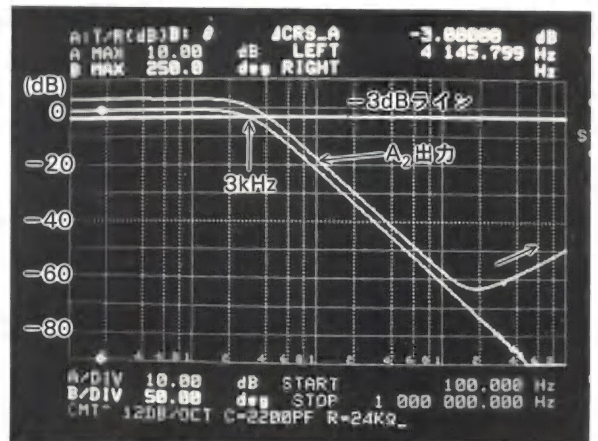
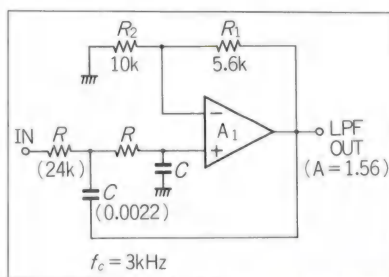
写真2は製作したLPFの減衰特性で，ほぼ計算どおりになっています。上側のカーブ（A_2出力）は通過域で1.56倍（3.86dB）の利得を持

〈写真2〉
3kHz LPFの
減衰特性

【図4】アンプA_2を省略する方法

【図5】標準的な18dB/octLPF

$f_c \fallingdotseq \frac{0.64}{2\pi CR}$

【図6】同一定数で構成したLPF

ち，200kHz以上から減衰特性が悪くなっています。これは使用したOPアンプの高周波特性が良くないためで，A_1より出力する場合は問題ありません。

A_1，A_2はデュアルOPアンプ(TL072)を使用しましたが，1回路で行うには**図4**の方法でも可能です。しかし，通過域で1.56倍の電圧利得を持つので，ほかでアッテネータを入れる必要を生じます。

18dB/oct LPF

12dB/octの減衰特性では不満な場合は抵抗，コンデンサをおのおの1個追加して18dB/octとすることができます。

標準的な方式としては**図5**のように，パッシブ・フィルタ(6dB/oct)と前に述べた12dB/octアクティブフィルタを組み合わせますが，そのままでは遮断周波数f_Cで6dB減衰し，f_C付近の減衰特性がブロードになってしまいます。

そこで，12dB/oct LPFのQを大きく($Q=1$)し，全体で−3dBポイントがf_Cになるようにします。

Qの設定はコンデンサC_2，C_3をアンバランスにする方法と，同一定数とするためアンプA_2の利得AをA＝2とする方法があります。

次に紹介する18dB/oct LPFは，変則的な設計をしなければなりませんが，18dB/octでも同一定数でフィルタを作れます。

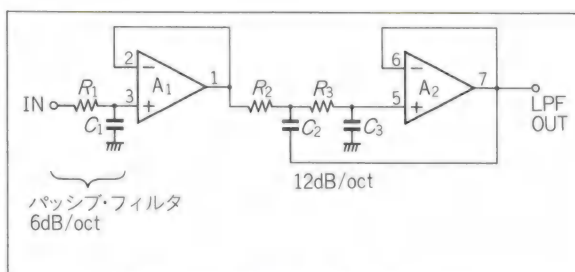
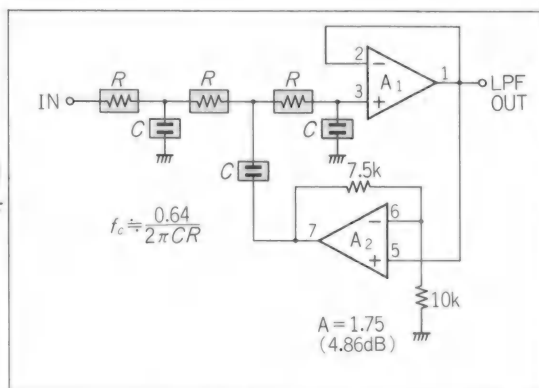



図6は，3段のCR回路網に正帰還をかける方法でたいへん便利なフィルタです。遮断周波数f_Cを求める係数(約0.64)は，実験的に決定しました。

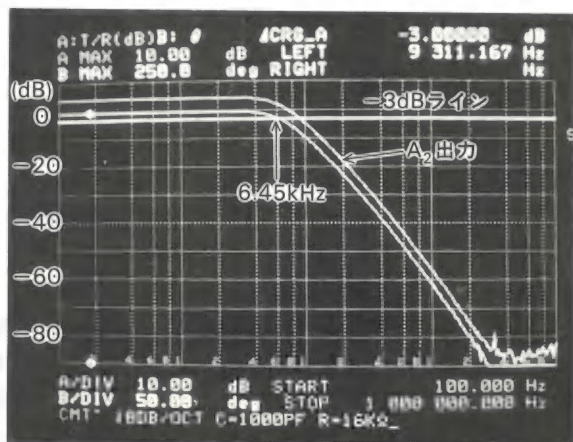
方法は3個の抵抗Rを同一定数の16kΩ，コンデンサCをすべて1,000pF(一般の$f_C=1/2\pi CR$は10 kHz)として，アンプA_2の利得を可変しながら，周波数特性が最平坦となる値を求めます。

利得Aは1.75より大きいと，f_C付近にピークを生じます。

写真3は最平坦な特性が得られたときの減衰特性で，−3dBポイントの周波数f_Cが約6.45kHzに低下しています。上側のカーブはアンプA_2出力の特性で，利得が約5dB大きくなっています。

遮断周波数f_Cが$f_C \fallingdotseq 0.64/2\pi CR$になるので，回路定数を決めるときコンデンサ$C$を都合の良い値として，$R=0.64/2\pi f_C \cdot C$を計算します。中途な値となったら，抵抗器を2本直列にすれば，正確なフィルタを製作できます。

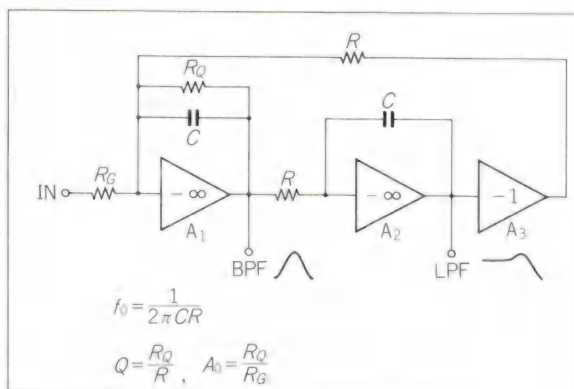
この回路も先に述べたように，

〈写真3〉18dB/octLPFの減衰特性

OPアンプ1回路でも実現できますが，通過域での利得が1.75倍になってしまいます。

すべてを反転アンプで構成したバイクワッド型フィルタ

今まで述べたアクティブ・フィルタは，能動素子としてOPアンプ・バッファか利得を持った回路を帰還ループに入れましたが，状態変数型と呼ばれる汎用性の高いフィルタ回路では，加減算回路，積分器でフィルタ回路を実現します。

その中でもバイクワッド型フィルタは，**図7**のようにすべてが反転アンプで構成(積分器も含む)されています。

【図7】バイクワッド・フィルタの基本回路

アンプA_1は抵抗R_Qが無ければ単なる反転積分回路ですが，コンデンサCと並列接続して不完全積分を行っています。

アンプA_2は真の積分回路で，利得が−1となる周波数fは，

$f=1/2\pi CR$

のときです。

アンプA_3は単なる位相反転の目的に使用され，その出力は入力に帰還してループを構成し安定な回

アクティブ・フィルタで使用する部品

●OPアンプ

アクティブ・フィルタで扱う最高周波数において、十分な負帰還量が得られる周波数特性を必要とします。DCから50kHz程度なら一般的なOPアンプを使用できます。

しかし、汎用OPアンプと言っても4558タイプよりも、交流特性が良く、入力抵抗の高いBi-FET型のほうが、使用する抵抗値の制約が少ない（入力バイアス電流が極めて小さい）ので、設計の自由度があります。

代表的な製品としては、TI社のTLシリーズ（TL-072、TL-082など）やNS社のLFシリーズ（LF-353、LF-412など）があります。

高周波でアクティブ・フィルタを実現しようとすると、当然、高速OPアンプを使用することになりますが、一般的には入力抵抗が低く、バイアス電流が極めて大きいので注意が必要です。

●コンデンサ

精度が良く、安定度の優れたコンデンサは入手も難しく、高価ですが、普通の用途ならマイラフィルム型で十分です。0.01μF以下ならスチロール型が良く、特性も良く、入手しやすい精度は±5％誤差です。

高精度を要求する精密なフィルタでは、双信電気のQSコンデンサをお勧めします。

高誘電率系のセラミック（パスコンなどで使用）や、タンタルコンは使用しないでください。

通常使用する容量を1,000pF、0.01μF、0.1μFといった値で設計すると、部品の標準化ができ経済的です。

●抵抗器

アクティブ・フィルタの遮断周波数を決定する要素のほとんどはCR回路網の時定数で、計算した定数どおりの部品を使用すれば、ほとんど無調整で製作できます。

コンデンサの容量は都合のよい値を選ぶ設計方法としたので、抵抗器の値が、中途半端な抵抗値となってしまうが、基板を設計する際に2本の抵抗器を直列接続できるようにしておくと便利です。

例えば、計算結果が122kΩだとすると100kΩと22kΩを直列にします。

一般的な抵抗器はカーボン型ですが、精度、安定度を要求する用途では±1％誤差の金属被膜抵抗器がよく使われています。

同一定数で、所定のQを設定するため、バッファ・アンプに電圧利得を持たせる設計をしているので帰還回路で使用する抵抗器は、高精度なものを必要とします。ここで使用する抵抗器は、必ず金属被膜型にしてください。

一般に高いQを設定するフィルタほど、コンデンサ、抵抗器は高精度な品種を選定する必要があります。

路とします。

この回路の特徴は，すべて反転アンプで構成されているのと，Qの設定が容易で，大きなQを必要とするバンド・パス・フィルタ(以下BPFと略す)で有利です。

回路構成は複雑ですが，特性の良いフィルタを作るときには便利な回路で，設計の手順は極めて簡単に行えます。

遮断周波数f_Cはなじみの多い

$f_C=1/2\pi CR$

で算出でき，コンデンサCを都合のよい値にし，抵抗Rを逆算して

$R=1/2\pi f_C\cdot C$

として求めます。

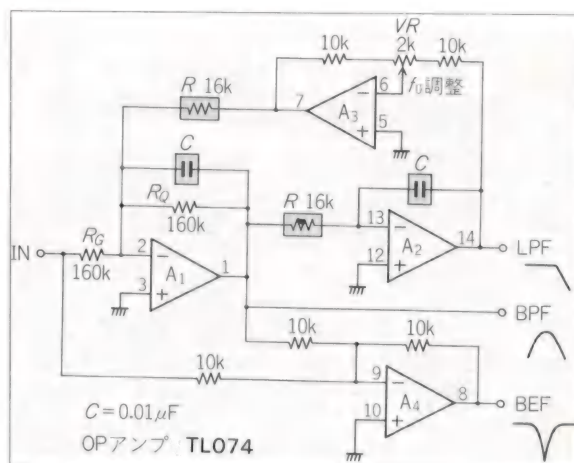
次にフィルタのQを設定するわけですが，12dB/oct LPFで使用する場合は，$Q=0.707$，BPFで帯域幅Bを狭くしたいときは，Qを大きく($Q=f_0/B$)設定しますが，ここでの実験では，$Q=10$とします。

Qの設定は抵抗R_Qで行い，遮断周波数を決定する抵抗Rとの倍率($Q=R_Q/R \therefore R_Q=Q\cdot R$)ですから暗算でも求められます。

フィルタ回路の利得Aは抵抗R_Gで決定しますが，LPF出力ではf_Cよりずっと低い周波数の通過域利得は$A_{LP}=R/R_G$，f_C(ピークの周波数)では$A_C=R_Q/R_G$となります。

一方，BPF出力を使用する場合は，中心周波数f_0での利得を言い$A_0=R_Q/R_G$として計算します。

このフィルタは，以上で述べたようにフィルタ回路を設計する際に必要な要素を単独に計算することができるため，フィルタの基本ブロックとして便利です。

しかし欠点として，たかが12dB/octのフィルタを実現するのにOPアンプが3回路必要です。このままでは，ハイパス出力やバンド・エリミネート(BEF)出力が得られないので，さらにOPアンプをもう1回路必要とします。

【図8】バイクワッド・フィルタ回路

バイクワッド・フィルタの実際

図8は，クワッドOPアンプ1個で作るバイクワッド型フィルタで，基本回路にアンプA_4を追加してBEF(一般にはノッチフィルタとも呼ばれる)を得ています。

この回路定数はf_0またはf_Cが1kHz，Qが10となっていることは先ほどの計算式からも理解できると思います。

反転アンプA_3に付加した可変抵抗器VRは，f_0またはf_Cを正確に合わせるために挿入してあります。これは，一般に高精度なコンデンサCが入手しにくい点を考慮したためで，無くてもかまいません。

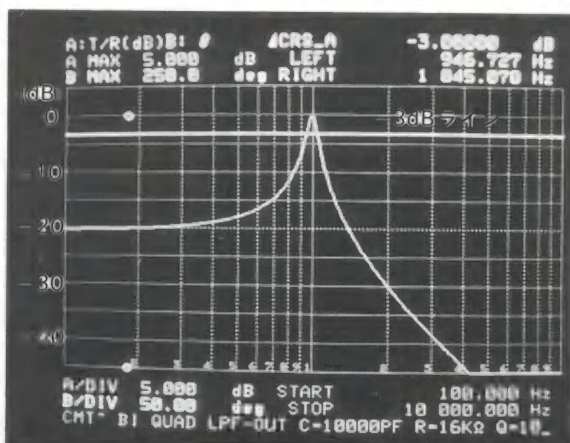
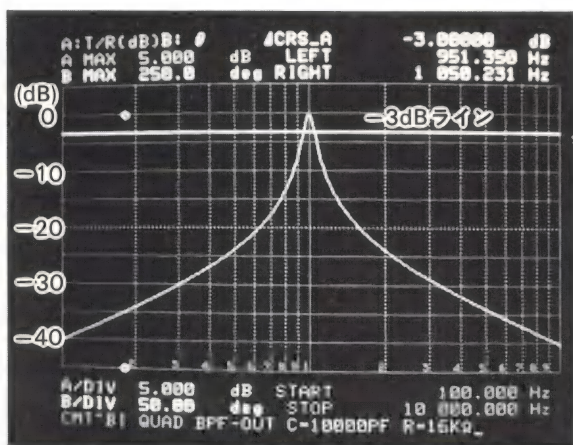
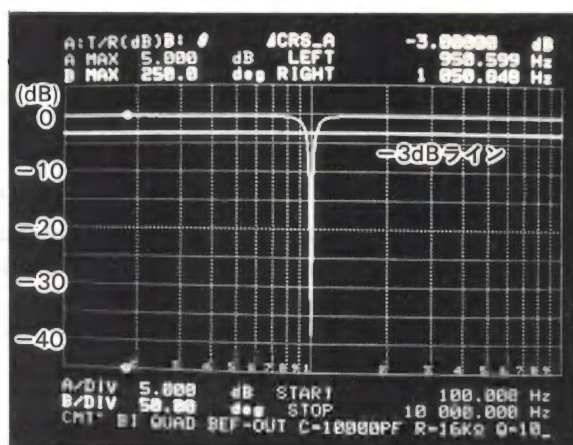
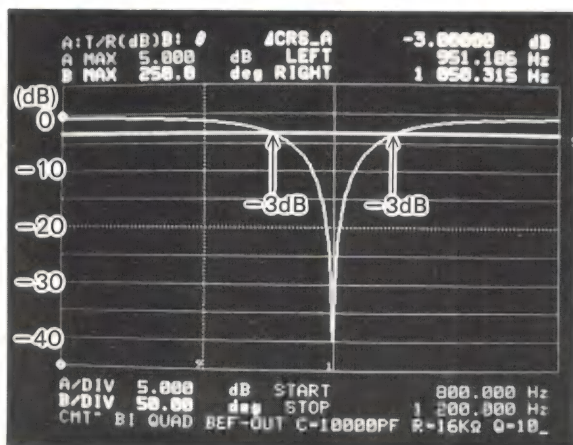
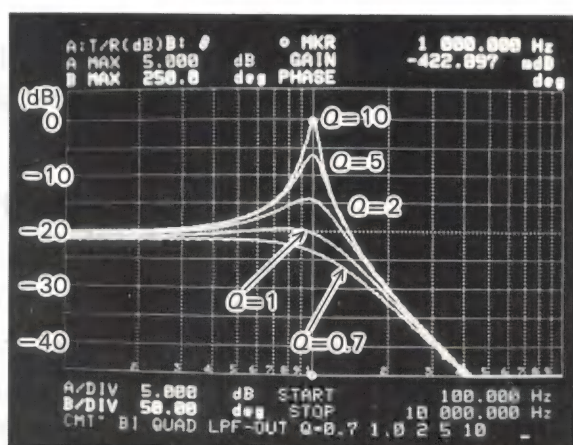
BEF(ノッチ)出力は，入力信号とバンド・パス出力BPFと加算することにより実現していますが，BPF出力は入力信号に対して位相が反転しているため，実際には引き算回路として動作します。

フィルタの特性

図8の回路定数でフィルタの減衰特性を測定してみました。

写真4はLPF出力の減衰特性で，遮断周波数f_Cにて大きなピークを生じていますが，これは$Q=10$に設定したためでf_Cよりずっと低い通過域周波数に比べてちょうど10倍(20dB)アップしています。

LPFは通常は高Qで使いません

〈写真4〉LPF出力の減衰特性 $f=100\text{Hz}\sim10\text{kHz}$

〈写真５〉BPF出力の減衰特性 f=100Hz～10kHz

〈写真６〉BEF出力の減衰特性 f=100Hz～10kHz

〈写真７〉写真６を拡大してみた f=800Hz～1.2kHz

〈写真８〉Qを変えると鋭さが変化する

から，通過域の利得を０dBとするには，$R_G=R$として設計するとよいでしょう。

バイクワッド型フィルタの最適な使い方は，バンド・パス・フィルタです。**写真５**はBPF出力の減衰特性で，中心周波数f_0が１kHzで，3dBダウンの周波数がおのおの951Hz，1,050Hzですから101Hzということになり，これよりフィルタのQを計算すると，

$$Q=f_0/\mathrm{B}=1000/101=9.9$$

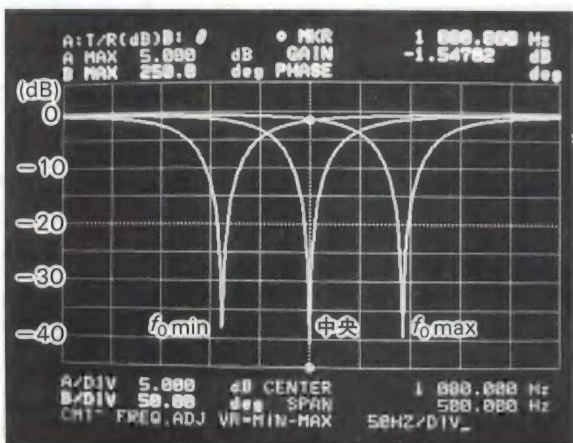
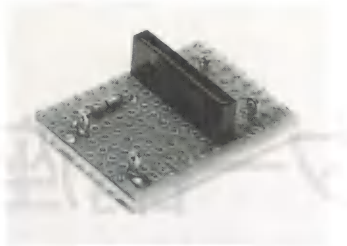
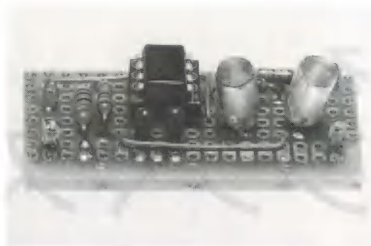
で，設計値にかなり近い値です。

バイクワッド型の応用としては２本の抵抗Rを連動して変化させることにより，中心周波数可変フィルタを作ることができます。

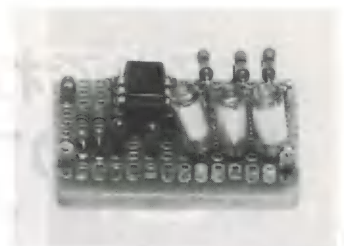
〈写真９〉中心周波数の可変範囲

写真６はBEF出力の減衰特性で，中心周波数f_0（１kHz）で大きな減衰量が得られ，これ以外の周波数ではほとんど減衰していません。

減衰カーブを見るとかなり急峻になっているので，拡大してみたのが**写真７**です。ノッチ・フィルタのシャープさもQで表現し，通過域より3dB低下する周波数（バンド幅B）の逆数に中心周波数f_0を掛けた値（$Q=f_0/\mathrm{B}$）がBEFフィルタのQになります。

当然なことですが，−3dB周波数を測定するとおのおの951Hz，1,050Hzで，BPFの帯域幅と同じ

〈写真10〉 CRフィルタ

〈写真11〉 12dB/octLPF

〈写真12〉 18dB/oct

〈写真13〉 バイクワッド・フィルタ

値になっています。

BEF出力は，ある特定の周波数のみ鋭く減衰できるので，例えば信号中に含まれる電源周波数(50, 60Hzのハム雑音)など除去に応用できます。応用上の注意事項は，むやみにQを大きく設計しないことで，使用するコンデンサの精度，安定度が悪いと中心周波数が変動して，大きな減衰量を得られなくなります。

フィルタのQ

写真8は，フィルタのQを0.7～10まで変更した場合の減衰特性で，$Q=0.7$がバタワース応答の減衰特性で，$R_Q=RQ=11.2\text{k}\Omega$とします。

$Q=1$は遮断周波数f_Cにおいて，利得Aが1になっていることがわかり，逆な言い方をするとQとはf_Cにおける利得と同じです。Qを2，5，10を大きく設定することにより鋭くなっています。

遮断周波数の微調整

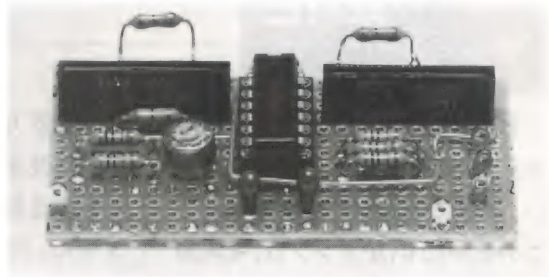
フィルタの遮断周波数を決定する抵抗器は，誤差±1%のものでも容易に入手できますが，コンデンサはそうもゆきません。

一般的に多く使われているマイラ・フィルム型は，誤差±10%が多くスチロール型で±5%です。もし±1%の精度を要求すると非常に高価で，入手も面倒です。

そこで，多少精度の悪いコンデンサを使用しても中心周波数を正確に合わせられるよう反転アンプA_3の利得Aを0.833～1.2倍まで可変できるよう可変抵抗器($VR=2\text{k}\Omega$)を入れました。

写真9は，可変抵抗器の中央に対して，左または右に回しきったときの中心周波数の変化で，約±8%ぐらい可変できました。

実際にはもう少し可変範囲を狭く($VR=1\text{k}\Omega$)したほうがよいかもしれません。

以上の実験基板を，**写真10～13**に示しておきます。

これでわかる OPアンプマスター講座

最終回 応用編2

窪田 登司

本連載講座も最終回です。前回の続きで応用編として，今回は，いろいろな発振回路を解説します。このほかにもOPアンプは，電源回路や高周波の整流，オーディオアンプ，各種センサー回路など多方面に応用されており，本誌にもたびたび登場しています。今後もますますOPアンプの応用は多くなることでしょう。

ところで第1回からのまとめとして，OPアンプの主なパラメータの測定ができるOPアンプテスターが，三田無線研究所(ブランド名『デリカ』)から発売されているので，これを使っていろいろのOPアンプの評価をしてみました。最終回のしめくくりとしてご紹介しておきましょう。

OPアンプ応用発振回路

OPアンプを応用した発振回路はいろんなバリエーションが考えられ，千差万別です。しかし，基本的には発振のためのルールがあり，それを逸脱するような適用はできません。たとえば，使用するOPアンプの利得帯域幅積を越えるような発振周波数は得られませんし，正弦波発振器の場合は波形のひずみも考慮するともっと低い周波数止まりになることは否めない事実です。

さて，正弦波の発振のためのルールですが，基本的には2つの条件が必要です。

①発振周波数の位相が入出力間で360度，もしくはその整数倍になること。

②発振周波数でのゲインが1以上であること。

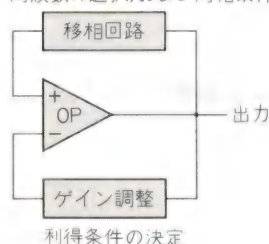
これらを実現するにはOPアンプの場合，正相，逆相の2つの入力端子を持っているので，**図1**のような構成で可能です。入出力を同相にする移相回路と利得条件を決定するゲイン調整回路を組み入れればよいわけです。

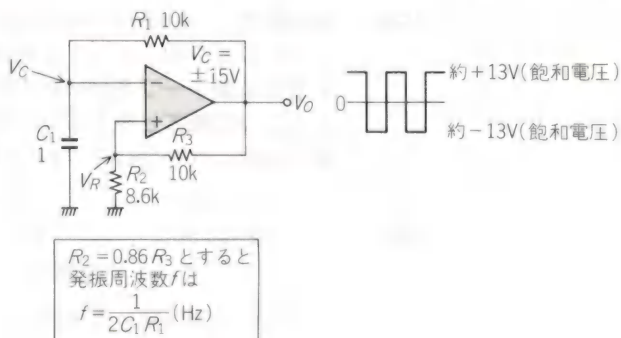
発振波形は用途に応じて正弦波や方形波，三角波，のこぎり波などいろいろありますが，工夫することによっていろんな波形が得られます。

それでは初めに最も単純なコンデンサの充放電を利用した方形波の発振回路からみてみましょう。

方形波の発振

コンデンサの充放電を利用してコンパレータ出力をオン／オフすれば方形波発振器となります。いわゆる非安定マルチバイブレータです(**図2**)。

原理は簡単です。いま，電源投入時にはコンデンサの両端電圧は0だとしますと，高利得OPアンプはノイズなどによって，＋側か－側か，どちらかに飽和します。いま＋側の飽和電圧出力(約13V～14V)

【図1】OPアンプを使用したCR発振器の基本形

【図2】方形波の発振

になったとしましょう。そうすると，コンデンサにそのV_Oが充電されるので，V_Cは徐々に上昇していきます。

一方V_Rは

$$V_R = V_O \frac{R_2}{R_2+R_3}$$

ですから，V_Cはこの電圧になるまで上昇し，これを越えるとコンパレータ動作が反転し，出力は－側の飽和電圧（－13V～－14V）になります。このとき，V_Rは

$$V_R = -V_O \frac{R_2}{R_2+R_3}$$

ですから，V_Cはその電圧になるまで反対の極性に充電されます。そしてそれを越えた時点で，また出力は反転します。

これを繰り返して方形波出力が得られます。

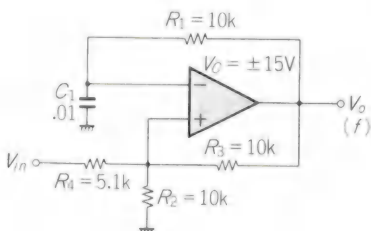
発振周波数 f は

$$f = \frac{1}{2C_1R_1\ln(1+2R_2/R_3)}$$

で与えられるので，$R_2=0.86R_3$とすれば$\ln(1+2R_2/R_3)=1$となるので，

$$f = \frac{1}{2C_1R_1}$$

となります。この例では$R_1=10$ kΩ，$C_1=1\mu$Fなので，発振周波数は約50Hzです。

R_1とC_1の組み合わせによって，1Hz程度の非常に低い周波数から，OPアンプにもよりますが数100kHzまでの発振器が可能です。

三角波の発振

コンパレータと積分器を組み合わせると，方形波および三角波の発振器となります。**図3**です。方形波を積分すると三角波になりますが，まさにそれを利用したものです。

動作は次のようになります。いまコンパレータOP_1出力が－側に飽和しているとします。そうすると，この電圧は積分回路OP_2によって反転し，V_Oはプラスの方向に上昇します。

一方，V_RはV_SとV_O間の電圧をR_2とR_3で分圧したものですから，V_Oが上昇するとV_Rも上昇します。したがってV_Rが0Vを越えるとV_Sは反転して＋側の飽和電圧になります。そうするとそのV_Sは積分されて（C_1に充電されるので）V_Oは下降し始めます。当然V_Rも下降し，0Vを過ぎると，コンパレータ出力V_Sは＋側に反転します。

以上のようにしてOP_1の出力は方形波，OP_2の出力は上昇，下降，上昇，下降を繰り返す三角波を発生することになります。

V_OとV_Sの波高値の間には

$$V_O = V_S\frac{R_3}{R_2}$$

の関係があるので，図のように100kΩにすれば，三角波と方形波の波高値は等しくなります。

電圧制御発振器VCO

方形波の発振で繰り返し周期を決めるのはR_1C_1の時定数によって決まるV_Cとリファレンス電圧V_Rがあることは図2でわかったと思います。そこでV_Rを変えて発振周波数を変えるようにしたものが，**図4**に示す電圧制御発振器VCO（Voltage Controlled Oscillator）です。

電圧を変えるとどのように充放電特性が変わるかを計算すると図に示したような発振周波数になります。直流電圧V_{IN}によって変化することがわかります。0V～±15Vで周波数は数10%変化します。

発振周波数を常に安定にするために，出力を整流し直流にして帰還させる方法や，文字どおり電圧によって周波数を変化させる回路によく使用されます。

【図3】
三角波発振回路

【図4】
電圧制御発振器

発振周波数

$$f = \frac{1}{C_1R_1\left\{\ell n\frac{(R_2R_3+2R_2R_4+R_3R_4)^2V_O{}^2-(R_2R_3)^2V_{in}{}^2}{(R_2R_3+R_3R_4)^2V_O{}^2-(R_2R_3)^2V_{in}{}^2}\right\}}$$

もしV_{in}とR_4がなければ，通常の方形波発振器となる

$$f = \frac{1}{2C_1R_1\,\ell n(1+2R_2/R_3)}$$

正弦波発振器〜移相形

OPアンプが使用されるようになる以前から正弦波発振器としてはよく見かけるものの一つに移相形というものがあります。CRによって位相を180度ずらせて，さらにアンプで180度ずらせば，計360度ずれるので，その周波数のゲインが1以上になれば，発振することになります。

移相形にはCRの挿入の仕方でハイパス形とローパス形があります。

移相回路の伝達関数$G(\omega)$は

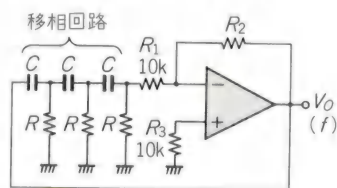
$$G(\omega)=\frac{1}{1-\frac{5}{(\omega CR)^2}-j\frac{1}{\omega CR}\left\{6-\frac{1}{(\omega CR)^2}\right\}}$$

と計算されるので，発振するためには虚数項が0でないといけないので，

$$6-\frac{1}{(\omega CR)^2}=0$$

から，

$$\omega=2\pi f=\frac{1}{\sqrt{6}\ CR}$$

ゆえに，発信周波数fは，

$$f=\frac{1}{2\pi\sqrt{6}\ CR}$$

です。

また，この関係を上式に代入して整理すれば，$G\fallingdotseq-1/29$となります。−というのは位相が180度遅れることを意味し，振幅は約1/29になることです。したがってOPアンプのゲインは発振周波数に対して29倍以上ないといけません。

1Hz以下の非常に低い周波数から数100kHzくらいまで安定に発振しますが，OPアンプによっては利得帯域幅積の小さいもので数100kHzでは29倍の利得が得られないものもあるので，高域限界はOPアンプによって決まります。

なお，波形ひずみの点からは，ローパス形の方が高調波が発生しにくいので有利です。

ウィーンブリッジ形発振器

ウィーンブリッジ形発振器は考案者の名をとってターマン発振回路ともいいます。**図6**のように移相回路を構成したもので，低ひずみで比較的安定，移相回路がシンプルで減衰量が少ないなどの特徴があります。移相回路がウィーンブリッジと呼ばれる交流ブリッジとよく似ているため，この名がありますが，ウィーンブリッジの平衡条件とは全く関係はないので，誤解のないようにしてください。

この伝達関数$G(\omega)$は，

$$G(\omega)=3+\frac{1/j\omega C_2}{R_1}+\frac{R_2}{1/j\omega C_1}$$

となります。いま$R_1=R_2=R$, $C_1=C_2=C$とすれば，

$$G(\omega)=3+j\left(\omega CR-\frac{1}{\omega CR}\right)$$

となるので，

$$\omega=\frac{1}{CR}$$

すなわち $f=1/2\pi CR$で，振幅が1/3となり位相は同相となります。したがってOPアンプの同相入力端子に，この移相回路を挿入し，ゲインを3倍に調整すれば，正帰還して発振することになります。

なお，CRが異なる時は，上式から発振周波数 fは，

$$f=\frac{1}{2\pi\sqrt{C_1C_2R_1R_2}}$$

となります。

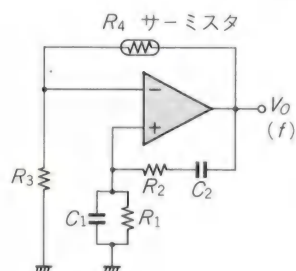
図6で増幅度を決めるのはR_4/R_3で，これが最終的に3倍になればよいので，電源投入時は増幅度が大きく，発振出力V_0が生じるにつれて増幅度を下げるように，この回路ではR_4にサーミスタ(例：STB-120V)を使用しています。

R_3に白熱電球(例：12V/10mA〜24V/5mA程度，その場合R_4は2k〜10kΩ程度に選ぶ)を使用することもあります。タングステン電球は電流が増加すると，抵抗が大きくなる性質があるので，自動利

ハイパス形

ローパス形

いずれも発振周波数は $f=\frac{1}{2\pi\sqrt{6}\ CR}$

【図5】移相形発振回路

$$f=\frac{1}{2\pi\sqrt{C_1C_2R_1R_2}}=\frac{1}{2\pi CR}$$

$\left(\text{ただし}C_1=C_2=C,\ R_1=R_2=R\right)$

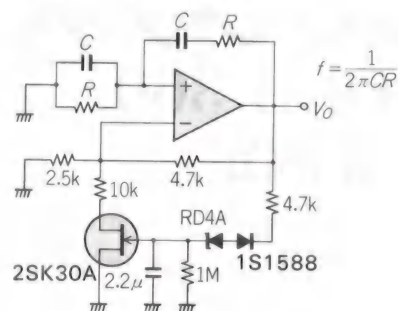
【図6】ウィーンブリッジ形発振回路

得調整回路(AGC)として使用できるわけです。

もっと高安定な振幅特性を得る場合はFETのチャネル抵抗を可変させて自動利得調整する方法があります。**図7**にその代表例を示します。

よく知られた接合形FETである2SK30Aを使用しています。このFETはゼロバイアス時は，チャネル抵抗は約1kΩ程度なので，直列抵抗10kとで約11kΩなので2.5kとの並列で，約2kΩがR_3ということになります。したがってゲインは

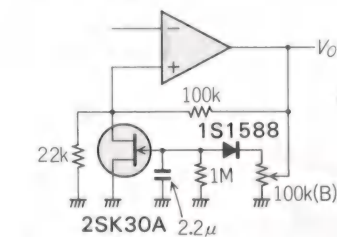
$$1+\frac{R_4}{R_3}\fallingdotseq 3.35倍$$

となり，容易に発振します。そして発振出力V_Oのピーク値が（ツェナー電圧約4V(RD4A)＋ダイオード1S1588のドロップ電圧約0.5V）≒4.5Vを越えるとFETは負のバイアスがかかり，チャネル抵抗は大きくなります。こうして全体のゲインは下がり，ちょうど3倍になったところで安定になります。

【図7】AGC回路を追加したウィーンブリッジ発振器

その他いろいろとAGC回路は考えられますが，安定度やひずみの点からは，このFETを用いたものが最も推奨できるものです。

振幅をボリュームで可変できるようにして調整をより完璧にしたものに**図8**のような回路があるので紹介しておきます(「オペアンプ回路の手ほどき」白土義男著NHK出版より)。

OPアンプの選び方ですが，まず発振周波数をいくらにするかによります。希望発振周波数のゲインが最低でも3倍は必要で，実際にはAGCのための余裕や低ひずみ化のためのゲインも必要なので3倍というわけにはいかないでしょう。もっと20倍とか30倍，あるいはそれ以上欲しいところです。したがって741Cでは10kHzくらいまで，LF356で数10kHzというところでしょうか。筆者はCA3140で500kHzを安定に低ひずみに発振させたことがあります。

また，OPアンプ自身の位相遅れも顕著になるので，計算したような発振周波数が得られないようにもなります。

【図8】自動振幅調整(AGC)回路の例

次にCRの選び方ですが，コンデンサは損失の小さいフィルムコンデンサにします。たとえば4.7μFと10kΩで，約3.3Hzを得る場合コンデンサは電解コンデンサでは不安定になる場合があります。フィルムにするべきです。

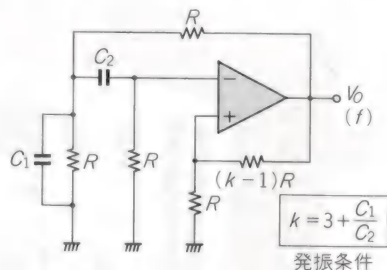
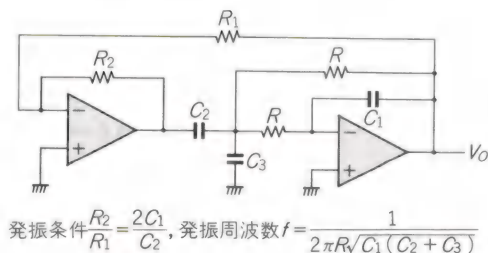
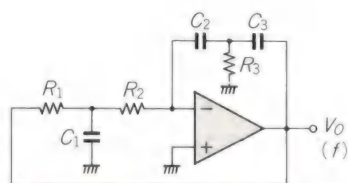
しかし，フィルムコンデンサの容量はせいぜい10μFまでで，それ以上は非常に大きくなるので，おのずと低域側も制限が生じます。

抵抗値の上限は使用するOPアンプの入力バイアス電流によって制限を受けます。Bi MOS形では10MΩを使用しているものを見かけたことがありますが，Bi FET形(LF356タイプ)では1MΩ，バイポーラタイプの741Cでは100kΩどまりとしたほうがよいでしょう。

その他の発振回路

OPアンプとCRを組み合わせた発振器は，先に述べたようにいろんなバリエーションが考えられます。

めずらしい発振器として**図9**に発振条件にかからないC_3で周波数を可変できる回路を示します。同様に図10も安定な振幅が得られるものとして知られています。

図11には少し発振条件が厳しい

【図9】その他の発振回路例①

【図10】その他の発振回路例②

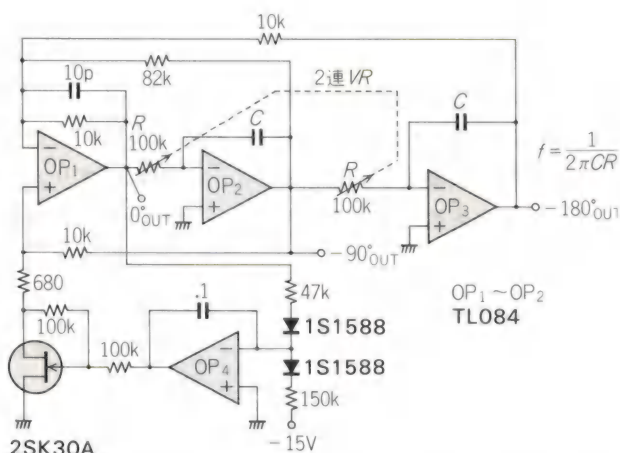
発振条件 $C_1 = C_2 + C_3,\ R_3 = \dfrac{R_1 R_2}{R_1 + R_2}$

発振周波数 $f = \dfrac{1}{2\pi\sqrt{R_1 R_2 C_2 C_3}}$

【図11】 その他の発振回路例③

【図12】 ステート・バリアブル・フィルタを応用した3相発振器

ですが，フィルタに帰還をかけて同相条件を満たした周波数で発振する回路を示します。

最後に**図12**にステート・バリアブル・フィルタに帰還をかけて発振させる3相発振器の例を示します。以前に本誌で(もう10年以上も前ですが)，超低ひずみ発振器と高性能ひずみ率計の製作記事を発表したことがありますが，そのとき製作した発振器です。積分回路で構成されているため，目的の発振周波数以外は凄い減衰をして高調波の発生がほとんどない，超低ひずみの発振器となります。容易に0.0001%(1kHz)のひずみ率が得られたことを覚えています。

原理は方形波を発振させて，これを積分回路によるフィルタを通してループさせ，高調波を減衰させたものと考えて差し支えありません。

図のように積分器が2段シリーズに接続されているので，90度づつ位相の異なった信号が得られます。発振周波数は

$$f = \frac{1}{2\pi CR}$$

で与えられます。

Rは100kΩ程度の2連ボリュームを使用すれば，周波数を10倍くらい容易に可変できます。またCをステップ可変にでもすれば，もっと広範囲の周波数の可変が可能となります。

出力の振幅制限回路は必要で，この回路は上述したFETのチャネル抵抗を利用したものです。OP_4は1S1588とともに整流回路を構成したもので，平滑性の良い，かつレスポンスの早い動作をします。

これは重宝！ OPアンプテスター

デリカ MODEL 360 による特性測定

〈写真1〉 OPアンプテスター「MODEL 360」(三田無線)と今回測定したOPアンプ類

三田無線研究所は昔からオシレータや真空管試験器，半導体試験器，インピーダンスブリッジなどの測定器のメーカーとして有名で，今回のOPアンプテスターも教育用だけでなく，実際の製造現場や試作・研究現場でも多く使用されている便利な測定器です(**写真1**)。

OPアンプのどういうパラメータが測定でき，どのようなスペックを持っているのか，**表1**に「MODEL 360」の取扱説明書から主な

規格を抜粋します。

測定値は最大1999カウントまでディジタル表示し，測定項目によりV，mV，nAの単位，および小数点が自動選択されます。だれでも読み違いのない正確な値が得られます。表示感度はノーマルと10倍感度の2レンジが装備されています。

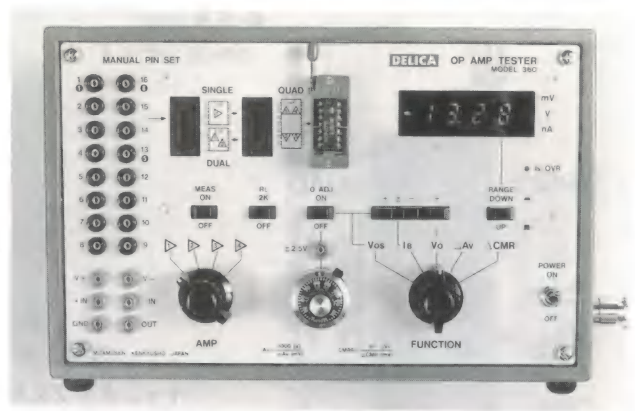
あとで述べるように最大出力電圧A_Vと同相分除去比CMRRは$\varDelta A_V$，$\varDelta$CMRから計算で求めるようになっています。写真**2**に「MODEL 360」の前面パネルを示します。

被測定OPアンプは，標準的なピン配列になっているものは，専用のソケットがパネル面にあり，その他のピン配列のOPアンプはパネル上のピンジャックを用いてマニュアルで接続します。キャンタイプの丸形OPアンプは付属のソケットがあるので，それを使用してパネル面のソケットに挿入できます(**写真3**(a)，(b))。

測定はすべて閉ループで行われるので，どんな素子でも簡単に良否テストも同時にできます。

測定時の負荷は無負荷または2 kΩの選択が可能で，負荷による特性の変化を調べるにも適しています。

被測定OPアンプは1度に1個だけ測定します。シングル，デュアルなどのOPアンプを同時に挿入しておいて次々に測定するというわけにはいきません。

また，測定回路の供給電源電圧は±15Vですが，外部電源をマニュアル接続して，±5V～±18Vで測定することも可能です(ただしA_VとCMRRは±12V～±18V)。**写真4**に測定項目の切り替えスイッチ部分のクローズアップを示します。

それでは，どういう測定ができるのか，実際に測定しながら，その原理も含めて(今までの復習を兼ねて)説明することにします。

〈写真2〉「MODEL 360」の正面パネル

① V_{OS}の測定

入力オフセット電圧です。理想的には入力に何も加えない場合，OPアンプの出力はゼロのはずですが，いくらか出ることがあります。これをオフセット電圧といいます。入力オフセット電圧の正確な定義

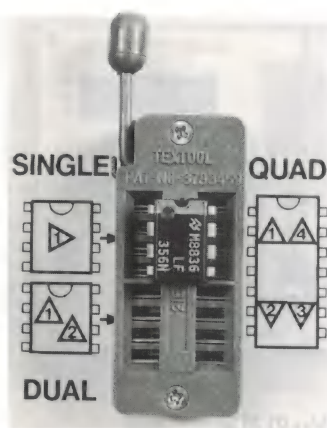
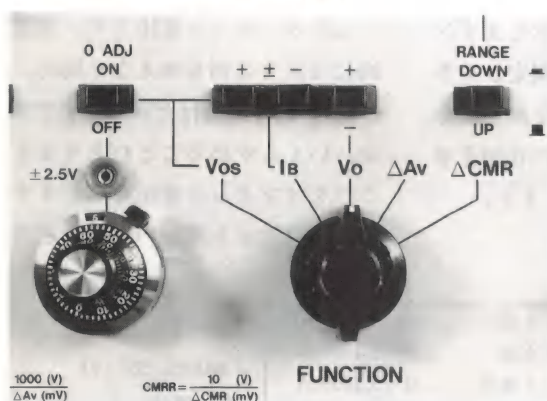
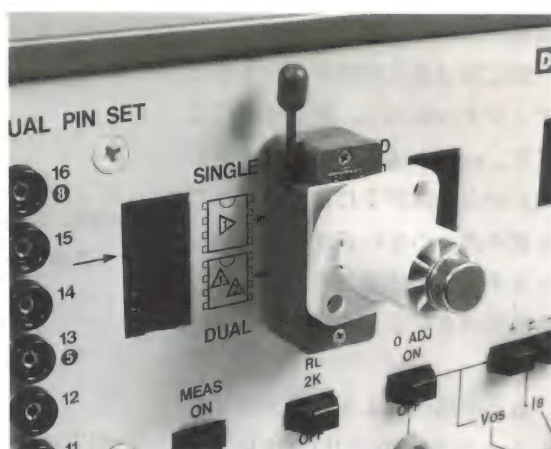
【表1】「MODEL 360」の主な規格

測定項目

V_{OS}	入力オフセット電圧	0～±19.99mV
$I_B(+)$	入力バイアス電流	0～±1999nA
I_{OS}	入力オフセット電流	0～±1999nA
$V_O(+)$	最大出力電圧	0～±19.99V
ΔA_V	開ループ利得誤差電圧	0～±1999mV
ΔCMRR	同相信号誤差電圧	0～±19.99mV

計算による測定項目

A_V	開ループ利得　1000V/mV～0.5V/mV（120db～54dB相当）
CMRR	同相信号除去比　1000V/mV～0.5V/mV（120db～55dB相当）（ΔA_V, ΔCMRより算出）
レンジ切換	マニュアル　2レンジ
確　度	V_{OS}, I_B, I_{OS}　±3%±2digit V_O　±2%±1digit A_V, CMRR　±10%±10dB
電源電圧	±15V±2%
過負荷警告	電源電流±150mA以上で警告ランプ点灯，電源電圧低下開始
出力電圧零調整	入力換算で±25mVまで可，10回転ヘリカルポテンショメータ使用
測定周波数	A_V, CMRR 測定信号　29Hz±10%
出力負荷切換	2kΩ 内蔵，外部負荷接続可能
ピン配列選択	(1) 専用ソケット（デュアルインライン） シングル，1回路型（TL071, 741 TYPE） デュアル，2回路型（TL072, 358 TYPE） クワッド，4回路型（TL074, 3403 TYPE） (2) パネル面上ピンジャックで任意のピン配列選択(最大16ピン)
ピンジャック	マニュアルピン配列選択用16個およびV+, V−, +IN, −IN, OUT, GNDの6個，±2.5V 1mA出力用1個
消費電力	AC 100V　50/60Hz　6VA
附属品	取扱説明書1部，16ピンレバーソケット，8ピン丸型ソケットアダプター各1個，ピンチップコード 5本，ピン－わに口コード 3本
寸法・重量	230W×145H×130D(mm)　1.7kg

◀〈写真3(a)〉
OPA用ソケット

〈写真3(b)〉▶
キャンタイプ丸形
OPA用ソケット

〈写真4〉
「MODEL 360」の測定項目切り替えスイッチ部のクローズアップ

【図13】 V_{OS}の測定

パラメータ／品種	V_{OS}〔mV〕	I_{B+}	I_{B-}	I_{OS}〔nA〕	V_{O+}	V_{O-}〔V〕	Av〔dB〕	CMRR〔dB〕
741C①	0.22	32	−33	0.5	14.2	−12.5	87	120
②	0.32	16	−16	0.4	14.2	−12.9	88	128
③	0.011	30	−30	0.5	14.2	−12.6	86	124
356N①	2.16	0.5	−0.5	0.6	13.5	−13.6	106	88
②	0.93	0.3	−0.3	0.2	13.5	−13.6	105	103
③	1.17	0.4	−0.4	0.3	13.5	−13.6	105	98
356H	0.11	0.5	−0.5	0.6	13.5	−13.6	104	87
308AN	0.01	3.6	−2.6	0.4	13.7	−13.5	107	104
CA3140E	0.42	2.2	−1.9	2.1	14.9	−12.9	96	109
TL071CP	1.43	7.2	−7.2	7.3	13.9	−13.3	100	96
TL072CP①	0.47	2.2	−2.3	2.1	13.4	−13.9	102	94
②	0.28	1.3	−1.2	1.4	13.9	−13.4	102	94
TL081CP	6.76	33	−33	33	13.3	−13.9	99	86
TL082CP①	2.24	11	−11	11	13.4	−13.9	102	97
②	0.98	4.9	−4.8	4.7	13.4	−13.9	102	110
TL074CN①	4.35	21	−21	21	13.3	−13.8	100	91
②	0.4	2.0	−2.0	2.0	13.3	−13.8	100	91
③	0.2	1.0	−1.0	1.0	13.3	−13.8	100	100
④	0.003	0.1	−0.1	0.1	13.3	−13.8	100	91
TL084CN①	0.85	4.0	−4.0	4.0	13.2	−13.9	101	102
②	2.28	11.0	−11.0	11.0	13.2	−13.9	100	92
③	1.59	7.0	−7.0	7.0	13.2	−13.9	100	118
④	1.79	9.0	−9.0	9.0	13.2	−13.9	100	91

【表2】「MODEL 360」による各種OPアンプの実測データの例

はオフセット電圧をゼロにするように入力端子に加えるべき電圧がV_{OS}ですが，直接測定することはできないので，本機では**図13**のような回路で，出力に生じる電圧を読んで測定しています。

今回の各種のOPアンプの実測結果は**表2**（V_{OS}欄）のようになりました。かなりバラツキがあることがわかります。なお，多くのOPアンプを測定してみますと，時間とともに徐々にV_{OS}は小さくなるものもあります。つまり安定しているわけです。こういった時間による変動や温度に対する安定度を見るにも本機は便利です。

②I_Bの測定

入力バイアス電流です。理想的にはOPアンプの入力インピーダンスは無限大であって欲しいわけですが，実際には入力端子にわずかながら電流が流れます。そのため出力に生じる電圧をゼロにする入力電流としてI_Bが定義されていま

【図14】 I_B 入力バイアス電流の測定

【図15】 V_O 最大出力電圧の測定

す。

＋(プラス)入力側のバイアス電流をI_{B+}，－(マイナス)入力側のバイアス電流をI_{B-}とし，この2つの電流を加算したものを入力オフセット電流I_{OS}と定義されています。

本機は**図14**のような回路で測定しています。S_1開でI_{B+}，S_2開でI_{B-}，S_1とS_2共に開でI_{OS}が測定されます。

測定例を表2のI_B，I_{OS}欄に示します。値はもちろん小さい方が入力インピーダンスが大きいわけですが，本機のI_B測定は主としてバイポーラ形のOPアンプを対象として設計されているので，FET入力形では，ほとんどゼロになります。レンジを高感度側にすれば読めますが，最小ケタはノイズ等の影響も出てきます。

以上のV_{OS}，I_B，I_{OS}などは本機の測定範囲を越えるようだとそのOPアンプは不良として差し支えありません。

③V_Oの測定

最大出力電圧です。入力電圧と出力電圧はある程度までは比例関係にありますが，電源電圧で決まる飽和するところがあり，それが最大出力電圧です。

本機は**図15**のような回路になっていて，電源電圧は±15Vです。

S_3で1側にすると－(マイナス)側出力，2側にすると＋(プラス)側出力の最大値が得られます。

測定例を表2のV_O欄に示します。極端にプラス側とマイナス側の値が異なるようでは不良ですので，簡単にOPアンプの良否を判別できることにもなります。また正常なOPアンプでは大体±12V～±14V付近になります。

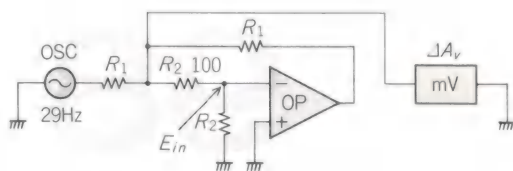
④A_Vの測定

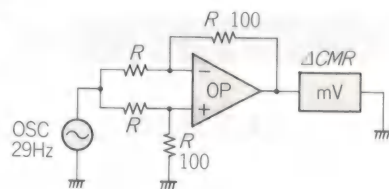
オープンループゲイン(開ループ利得)です。2つの入力端子間の電圧変化E_{in}に対する出力電圧の変化分E_Oの比を，開ループ利得または単に電圧利得と定義されています。単位はV/mVまたは，dBです。

本機は図**16**のような回路でE_{in}に比例する$\varDelta A_V$を測定し，

$$A_V = \frac{1000}{\varDelta A_V} \ [\mathrm{V/mV}]$$

から計算します。

一般にはA_Vは直流での値として発表されていますが，本機は測定の安定度を高めるために29Hzの交流で測定しています。

【図16】 A_V開ループ利得の測定

【図17】
CMRRの測定

その測定例を，表2のA_V欄に示します。

⑤CMRRの測定

同相分除去比です。OPアンプは差動アンプですので，2つの入力端子に同相信号が加えられると増幅されないで，出力はゼロになるのが理想的ですが，実際にはいくらかアンバランスがあり，増幅されて出力されます。CMRRは差動信号の利得A_Vと同相信号の利得A_Cの比として定義されています。

本機では，**図17**のような回路で同相信号による出力電圧$\varDelta$CMR(mVで表示)を測定し，

$$\mathrm{CMRR} = \frac{10}{\varDelta \mathrm{CMR}} \ [\mathrm{V/mV}]$$

から計算します。A_Vと同様に29Hzの低周波で測定されます。

この測定例を，表2のCMRR欄に示します。A_VとCMRRは直流で測定した値より小さい値になっていますが，それは交流信号29Hzで測定しているためですから，メーカー発表の特性表があれば，その周波数付近の値と比較することが肝要です。

98 VS マック

栗原 信義

最終回/データベースソフトの操作性(2)

自動実行機能

「ファイルメーカーPro」にはスクリプトとボタン機能というものがあります。

スクリプトは検索やソート(並び替え),印刷といったよく使う機能を登録しておき，メニューからの選択やショートカットキー(コマンドキーと数字キーなどの組み合わせ)で実行できるようにするものです。

ボタン機能は画面上に設けた枠(ボタン)をマウスでクリックするだけで，コマンドやスクリプトを実行できるようにする仕掛けです。

ボタンで実行できるのは，データベースを操作するのに必要なほとんどのコマンドと前述のスクリプトです。

この2つが実行できれば「ファイルメーカーPro」を立ち上げて必要なファイルを開くこと以外の通常の操作はすべて行えると思ってもよいでしょう。

ただし，これらの設定内容はユーザーに任されているものであり，正しく設定しなければならないことはいうまでもありません。

Macではマウスの利用が前提となっていますが，データベースソフトで画面にボタンを配置し，マウスのクリックで実行できるコマンドやスクリプトをユーザーが任意に設定できるということは，お仕着せソフトの枠を大きく超えるものとして評価できるでしょう。

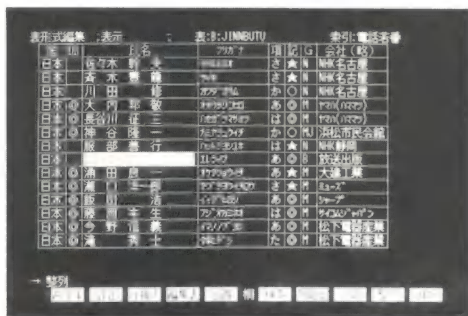
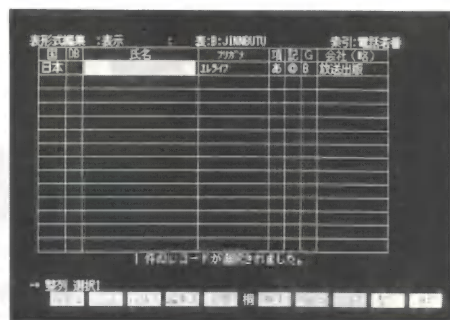
「桐」にはこうした機能はなく，通常はすべての操作をファンクションキーで機能を選択しながら対話形式で進めて行きます。

唯一可能な自動実行機能としては前回述べた一括処理のファイルを呼び出すことですが，これはあくまでもコマンドを記述した1つのプログラムを呼び出すことであり，「ファイルメーカーPro」のスクリプトとボタン機能に対応するものではありません。

【図1】「桐」vs「ファイルメーカーPro」の処理スピード比較

処理スピード

現在，筆者の手元には「桐」のデータとして1,132件を収めた住所録ファイルがあります。このデータをそのまま「ファイルメーカーPro」に移して両ソフトの処理スピードの違いを見て見ました。

今回対象とした比較項目は，主にデータベースを利用(参照)するときに行う処理を想定したものと

〈写真 1〉
検索画面

〈写真 2〉
選択画面

しました。

データを入力する場合はソフト側の処理スピードよりも，入力そのものに要する時間のほうがはるかに長いためスピードはあまり問題にならないからです。

図 1 が結果を示したものですが，全体的にかなり特徴的な傾向を示しているのがわかると思います。

結果を詳しく見ますと，まず処理条件としてデータの検索スピードを調査しました。

これは入力したデータの中から必要なものを探し出すというデータベースソフトにとってはもっとも特徴的な作業です。

ところで，この「検索」という作業では両ソフトでちょっとした考え方の違いがあります。

「桐」では「検索」と「選択」は明確に区別されていますが「ファイルメーカーPro」ではこの区別がありません。

「桐」で「検索」を行うと条件に当てはまる最初のデータが常に1件だけ選び出されます。

この場合，検索を行う項目に対して事前に整列を行っていると検索条件に合った最初のデータが画面に表示され，データにカーソルが当たります。そしてそのデータの下に連続するデータが並ぶという状況になります。

ただし，この処理では原則として必要とするデータに対して，完全に一致する条件を入力しないと目的とするデータを検索できませんので注意が必要です。

これに対して「選択」は入力した条件に一致するすべてのデータを探し出し，これを一覧表示します。

このとき，特定のデータ内容と完全に一致した条件を与えると，該当する1件のデータを選択して1件だけを表示します。

つまり全データの中の1件だけに該当するデータを与えた場合，「検索」では全体を表示している中の該当データにカーソルが移動する(**写真 1**)のに対して「選択」では該当データ1件だけが表示される(**写真 2**)という違いになって現れます。

これに対して「ファイルメーカーPro」には「桐」の「検索」コマンドに対応する処理はありません。

「検索」というコマンドで処理を行うと基本的にはすべての条件で「選択」となります。

つまり住所録で「東京」に住所がある人という条件を与えて検索すると，東京在住のすべての人が選択されます。

そして東京都渋谷区××○○5-3-20といったすべての住所を入力して検索をすれば，該当するデータ1件だけが表示されます。

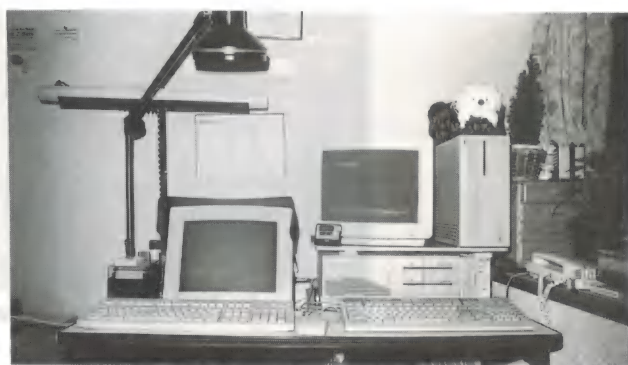
つまり「ファイルメーカーPro」では，全体が見えていてその中の1件にカーソルが当たってという検索状態はない，というところが「桐」とは違う部分です。

このため「ファイルメーカーPro」では検索を行うと必ずデータが絞られた状態になりますので，次にデータ全体から再び検索を行うためには選択状態を解除しておく必要があります。

さて図1に戻ると，この「選択」で1,132件の全データの中から「田中」という姓を持つ人だけを選び出す作業を行って見ました。

結果は11件のデータが選択されましたが，これがすべて表示されるまでの時間で「桐」が3.8秒，「ファイルメーカーPro」が2.4秒という結果になりました。ただし，これはいずれもソフトを立ち上げ，そのままの状態から処理を行ったものです。

さらに，選択を解除し，同じ作業を行って見たのが2つ目のデータです。ここでは「桐」が1.2秒，「ファイルメーカーPro」は同じ2.4秒という結果になりました。

「桐」は1回目と2回目で3倍のスピード差が出ました。これは「桐」の場合データがディスク上にあるのか，メモリー上にあるかの違いです。つまり1回目ではまだ全部のデータがメモリー上に展開され

〈写真3〉筆者宅のパソコン(左：Mac，右：98)

ておらず，ディスクから読み込みながら検索を行っているためです。

もしデータがフロッピーディスクにあると，読み出しスピードが桁違いに遅くなるため100件を超えるようなデータでは満足な処理スピードが得られません。

次に該当するデータが1件だけ，つまり特定のデータに完全に一致する検索条件を与えて処理を行って見ました。

ここでは「桐」が1回目と2回目でさらに大きな違いがでました。初回で6.7秒，2回目以降では1.1秒と約6倍の違いとなりました。

これに対して「ファイルメーカーPro」はどちらの場合でも1.3秒という結果でした。

次の高速検索という項目は「桐」にはなく，「ファイルメーカーPro」に用意されているコマンドです。これは「=，>，≧，<，≦，…，!，□」などの記号を使って数値あるいはテキストとして入力された数値データなどから，範囲検索を行う場合に利用します。

そして「ファイルメーカーPro」にはさらに順次検索というコマンドもあり，これはその名のとおりすべてのデータを順番に検索してテキストフィールドの部分一致検索を行います。

ただし，この処理では1,132件のデータを順番に検索するとそれだけで約17秒ほどかかります。

このため，氏名で検索したい場合，姓での検索，つまりデータの頭からの文字列で検索する場合は，通常検索で行うほうがはるかに早く目的のデータが得られます。

しかし，名のほうで検索したいといった場合には，順次検索の中に用意されている「==*""」といった書式を使って必要な文字列を""の中に入力して検索します。

「桐」の場合には，検索でも選択でも文字列を入力した後で「部分一致検索；1:しない，2:含む，3:含まない，4:先頭一致，5:末尾一致」の中から要求に合うものを選択するという方法です。

「桐」でも「選択」の場合，すべてのデータをサーチしなければなりませんが，2回目以降の作業の場合には約1秒程度，ほとんど瞬時という感覚で選択が行われます。

次にソート(並び変え)ですが，これもデータを大まかに見る，データに現れた傾向を読むといった場合には見方を変える，つまり切り口を変えるという意味で重要な処理になります。

図1の結果を見ると「桐」も「ファイルメーカーPro」も16秒ほどかかっています。しかし，「桐」のほうは2回目以降での処理時間は，約1秒程度となり「ファイルメーカーPro」は変わりません。

この差は，実は処理方法の違いが大きく影響しています。「桐」の場合，ソート条件を初めて登録して実行すると約16秒かかりますが，これで得られた結果はファイル上に書き込まれます。

2回目に同じ処理を行うと今度は，ソート結果をファイルから読み出してくるという動作になります。

そしてとりあえず画面に表示されるのは1画面分の20行程度だけです。このためほとんど瞬時にソートが終了したように見えますが，実際はファイルの読み込みということになります。

試しにデータをスクロールすると，次々にデータをディスクから読み出す作業が行われることから処理内容がファイルの読み出しであることがわかります。

ちなみに「桐」ではこうした処理を行っているためソート条件を多数作ったりした場合，実データの容量よりもかなり大きなファイル容量となっていることがあります。

ちなみに今回テストに使った住所録のテキストデータとしてのファイル容量は，200kバイト程度ですが，「桐」のデータベースファイルとしての容量は，1,200kバイトと約6倍となっています。

「ファイルメーカーPro」では「桐」のテキストデータを読み込んだ状態で約300kバイト程度でした。

さて，ここで圧倒的に差が出たスクロールについて考えて見まし

ょう。

「桐」の場合，罫線などはグラフィック画面を利用して表示していますが，データの内容そのものはすべてテキストデータとして画面表示が行われます。もちろんスキャナなどを使って取り込んだグラフィックスをデータとして読み込むと，グラフィック画面を使うわけですが，今回はこのテストは省きました。

NECのPC-9801は文字を専用のメモリー(漢字ROM)に持っていて，これを専用のテキスト画面に表示するという方法で高速な処理を行っていることについては今までにも述べてきましたが，この特徴がいかんなく発揮された結果がスクロールスピードの比較でしょう。

これに対して「ファイルメーカーPro」のほうは文字の1つ1つをディスク上のファイルから読み込み，これをグラフィック画面に展開するという作業が必要なため，スクロールは極めて苦手な処理となります。

スクロール中の画面を見ていると，1行分を約3回に分けて画面を描き直しながらスクロールが行われます。文字の読み込みと展開に加えて，さらにこの画面の再描画が行われるわけですから，全体として処理が重いことは十分に理解できますが，98を利用していてMacに移ると大量のテキストデータのスクロールでは，兎と亀の違いを実感してしまいます。

画面の再描画については，グラフィックのアクセレレータなどを利用すればかなり早くなるのでしょうが，コストもかなりかかります。

さて，処理スピードの比較の最後は，実際にデータベースを利用するためにコンピュータ上でソフトを立ち上げてファイル内容を表示してくれるまでの時間を計りました。

「桐」はDOS上のすべてのソフトがそうであるように，まず「桐」を立ち上げてから必要なファイルを読み込むという手順が必要です。このため18.5秒という時間の中にはリターンキーを4回押すという操作が入っています。

これに対して「ファイルメーカーPro」は必要なデータファイルをダブルクリックするだけです。

まずアプリケーションが起動し，次にファイルが読み込まれて立ち上がるまで，途中の操作は一切不要です。

実際の起動時間としては半分程度ですが，オペレーションとしてはこれ以上シンプルな方法はないでしょう。

このあたりが98とMacの基本的な違いを大きく感ずるところです。

おわりに

誌面が少なくなってきました。

さまざまなオペレーション環境などについても述べるつもりでしたが，基本的にデータベースソフトと言うのは極めて奥が深く，一通りの説明では理解できない部分も多いと思います。

筆者も当初は「桐」というソフトに惚れこんでいたため，これだけはMacの世界ではまだまだ対抗できないだろうと思っていたのですが，実際に使い込んでみるとむしろ総合的な利用環境では，Macのほうが上であると実感しています。

スクロール性能では圧倒的に劣りますが，1,000件を超すデータに対してスクロールで検索するということは現実的ではありません。

適切な検索方法などを身に付けることでむしろ総合的な処理時間では「ファィルメーカーPro」のほうが上回っているとも言えるでしょう。

ここでは，98とMac上で利用できる代表的なデータベースソフトの特徴を述べてきましたが，機能面での大きな違いは「桐」がグラフ機能を持つのに対して「ファイルメーカーPro」にはないこと，そして「桐」の一括処理コマンドがきわめて強力で，これを利用して特定の業務用ソフトを組むことによって付加価値をつけ，これが商売になるほどのものであることです。

そして「ファイルメーカーPro」のほうは，その優しいユーザーインタフェースとハイパーテキストと同じようなマウスによるオペレーションを構築することができることなどが大きな特徴となっています。

98の世界で作り上げたデータもそれがテキストデータであれば，簡単な操作で完全にMac上で利用することが可能です。

このシリーズは今回で終わりますが，筆者もこの1年で98からMacに，ほとんどの作業を移行してしまいました。

どちらが良いかは周囲との関係などもあり一概には言えませんが，少しでも98の世界とMacの世界を知る手がかりになれば幸いです。

テレビの受信障害対策に新技術
電波吸収体やSHF放送局が活躍

北嶌 秀博

テレビの受信障害は，平成３年度に電波障害防止協議会で受け付けた件数によると，全国で約11.6万件で，そのうちの約30％が建造物に起因しています。

建造物によるテレビ受信障害の対策方法としては，**図１**に示すように建造物側の対策と受信側の対策のほか，SHF放送局を建設する方法があります。

この中で最もオーソドックスな対策が受信側での共同受信施設の設置ですが，最近では，建造物側に新しい技術を使った対策や，SHF放送局の設置という新しい対策が取り入れられるようになりましたので紹介いたしましょう。

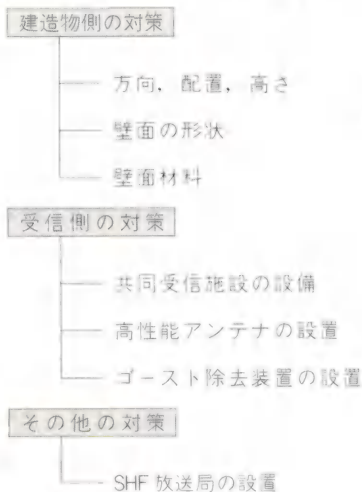
電波吸収体の登場

まず，壁面材料による対策です。

建造物側の対策
- 方向，配置，高さ
- 壁面の形状
- 壁面材料

受信側の対策
- 共同受信施設の設備
- 高性能アンテナの設置
- ゴースト除去装置の設置

その他の対策
- SHF 放送局の設置

【図１】
受信障害対策の方法

建造物による受信障害のうちしゃへい障害は，建造物が電波を通さない物質である限り日照の陰と同様になくすことはできません。

また，反射障害については建造物の壁面にテレビ電波がよく反射する材料を使用すると，障害範囲はかえって大きくなります。

表１に金属板を100％とした場合の主な材質の反射率を示しています。反射率100％とは材料にあたった電波が全部反射することです。

反射による障害範囲を狭くするためには，壁面材料に反射率が低いものを使用する必要があります。

ただし，ガラスや磁器タイルについては，素材自体の反射率は数％と低いのですが，透過した電波が建造物内の壁や金属など反射率が高い物質で反射され，再び建造物の外に反射波として出ていくため反射障害の改善効果は小さくなります。

そこで電波が物体内を通過するとき電波のエネルギーを熱エネルギーに変換し反射波を吸収する特性を持つ材料が開発されました。これが電波吸収体で，通称フェライトと呼ばれています。

米国のステルス戦闘機に使用されているレーダ反射塗料も電波吸収体の一つと言えるでしょう。

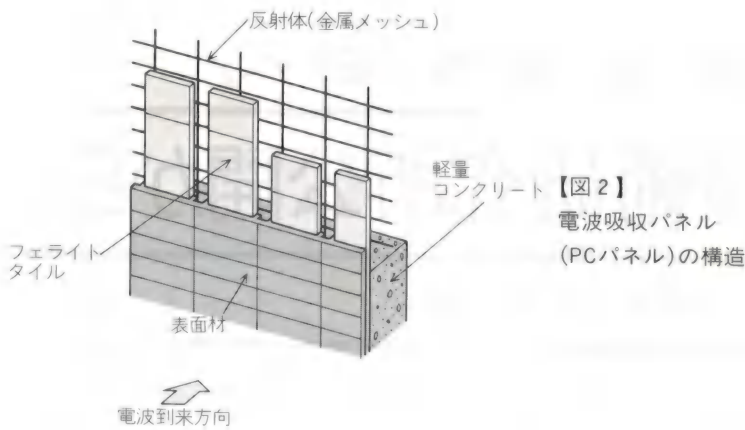
電波障害対策として使用されるフェライトは，酸化鉄と２価の金属酸化物を混合して焼付けしたもので比重が約５と比較的重く，約10×10×１(cm)のタイル状になっています。

以前は，フェライトを建造物の壁面に直接貼り付ける方法がとられていましたが，超高層化する現在の建造物では，工法上の問題などから**図２**に示すようにフェライトタイルを壁面内にあらかじめ埋め込んだPCパネルが使われ，電波吸収パネルと呼ばれています。フェライトのサンドイッチと言った

材　質	反射率(％)
金属板	100
鉄筋コンクリート	70
磁器タイル	3
ガラス	3
高性能熱線反射ガラス	80
フェライト	1

【表１】壁面用材質と電波反射率(例)

【図2】
電波吸収パネル
(PCパネル)の構造

〈写真1〉 個別受信の例

ところでしょうか。

電波吸収パネルのフェライトタイルの素材や厚み，各部材との間隔などは，反射障害改善の対象となる周波数や入射角度によって異なってきます。また，フェライトタイルが埋め込まれた電波吸収パネルを建物壁面の何処に，どの範囲に使用するかは，建設される建造物の立地条件等により異なるため，建造物ごとに電波吸収パネルの設計を行う必要があります。

表面材には反射率の低いものを使用する必要がありますが，磁器タイルや花こう岩など陶板や石板を用いたものが実用化され，意匠の点でもバリエーションが広がっています。

地上局用のSHFパラボラアンテナ

もう一つ，受信障害の対策として最近SHFテレビジョン放送による対策が注目されてきています。

実際に，ある地域で広範囲な受信障害が発生した場合には，SHFテレビジョン放送と共同受信施設など，どの対策方法を採用すれば経済的に有利かを総合的に検討する必要があります。

その選択の分岐点は，従来約3,000世帯と言われていましたが，最近はSHF関連機器の性能向上や低廉化により約2,000世帯が，その分岐点と考えられています。そして，なお一層の低廉化を目指した研究が進められています。

また，河川や道路など横断箇所が多い場合や，都市美観の問題などから電線類の地中埋設化地域が対象となる場合には，戸別に受信できるSHF放送局による対策が威力を発揮します。

この局の周波数帯域は，衛星放送などと同じ12GHz帯のSHFですが対策用SHFは衛星放送(BS)と衛星通信(CS)の中間に都市受信障害対策用としてチャネル番号63〜80までの18チャネルが割り当てられています。

わが国における最初のSHFによる受信障害の対策は，昭和54年6月(1979.6)，東京都足立区に設置されたSHF局によるものです。

当時の受信用パラボラアンテナは直径が40cm必要でしたが，最近民間企業体として初めて東京都渋谷区に設置されたSHFテレビジョン中継局の場合は，受信技術の進歩により30cmのパラボラアンテナで十分です(**写真1**)。

平成2年6月(1990.6)に放送法及び電波法の一部が改正され，放送事業者以外の建築主なども受信対策用SHFテレビジョン中継放送局の免許取得が可能となり，同年12月，その第一号が当社の施工により仙台市に設置されました。仙台市郊外の高さ100mの観音様の頭上に送信アンテナを設置しています。

このように電波障害対策も共同受信施設の設置が中心の時代からSHF放送局や電波吸収体が使用されるようになり，現在ではこのために設置されたSHFテレビジョン中継局が4局，電波吸収体が全国10数箇所の建造物で採用されるまでになりました。

放送設備の総合技術会社である私どもでは，受信障害対策に関する新技術の開発に取り組むと共に複雑化する電波障害の状況に応じたコンサルタント業務を実施しております。

(㈱NHKアイテック
受信・CATV事業本部)

「飛翔　放送新世紀へ」

NHK放送技術研究所公開から

葦名　保茂

NHK放送技術研究所の一般公開が5月28日(金)から30日(日)までの3日間，東京都世田谷区砧の当研究所で開催されました(**写真1**)。

この公開は，昭和5年6月の開所を記念して毎年行われているもので，一般の方々が研究所の成果をご覧いただける年1回の機会です。展示内容は番組制作，送信，受信など放送にかかわる技術全般にわたり，基礎研究から応用・開発まで広い範囲におよんでいます。

今回は30会場で45項目の展示が行われました。期間中，約1万2000人の人々が訪れ盛況でした。

ここでは，展示の中から主な項目を選んで紹介することにします。

新しい放送サービスのための研究

ハイビジョン壁掛けテレビ

現在1日8時間のハイビジョン試験放送が行われ，100万円前後のハイビジョン受信機が手に入るようになりました。このハイビジョン受信機の重量は，ブラウン管式の36インチクラスでまだ90kg前後あります。

ハイビジョンの迫力ある映像を十分に楽しむには大きな画面が理想的ですが，受信機の重量や容積を考えると，いわゆる「壁掛けテレビ」がハイビジョンのディスプレイとして最適です。

当研究所では，大型のディスプレイを比較的容易に製作できるプラズマディスプレイ(放電ディスプレイ)を選び，今世紀末を実用化の目標として研究を進めています。

今回展示した40インチのパネルは2枚のガラス基板をはり合わせた構造で，パネル製作技術の向上により発光部の開口率を高めるとともに，封入するガスの組成を最適化することにより，昨年展示したパネルより明るくなっています(**写真2**)。

今後は40型パネルの開発で得られた大型パネルの製作技術，駆動技術を土台にして，長寿命化をはじめ，高コントラスト化，省電力駆動などの研究開発を進め，「壁掛けテレビ」の実用化をはかっていきます。

〈写真1〉 入場者を迎えるNHK技研

〈写真2〉 40型ハイビジョン壁掛けテレビ

〈写真 3〉 ハイビジョンSuper-HARPハンディカメラ

〈写真 4〉 ハイビジョン方式変換装置

ハイビジョンSuper-HARP ハンディカメラ
～オールラウンドカメラを目指して～

夜行性動物の生態観察やオーロラなどの暗い被写体の撮影には高感度のカメラが必要です。ハイビジョン番組制作の多様化に対応可能な超高感度カメラの実現を目指して，HARP方式撮像管の研究を進めています。

昨年開発した1インチSS(電界集束・電界偏向)型Super-HARP管を用いた超高感度・高画質カメラは，ハイビジョン映像の世界を大きく広げ，すでに天体撮影などに活用されていますが，今回展示したSuper-HARPハンディカメラは，新たに開発した小型の2/3インチSS型撮影管を使って，機動性・運用性を改善しました(**写真 3**)。

昨年展示したスタジオ用標準カメラに比べ，大幅に小型・軽量化され，ハンディカメラとしてビデオロケなどに使用することができます。このカメラの最大感度は，標準型2/3インチサチコンカメラの約80倍(2000ルクス，F25相当)，最低被写体照度約9ルクス(F1.7，ノーマルゲイン)です。このカメラはターゲット電圧を変えることにより，真昼から月明かりまで極めて広い照度範囲で撮影することができます。

ハイビジョン方式変換装置

日本では走査線数1,125本，毎秒60フィールドの規格で毎日ハイビジョン試験放送が行われていますが，ヨーロッパでは走査線数，フィールド周波数とも日本と異なっています。

昨年，国際エレクトロニックシネマフェスティバルのドキュメンタリー部門でグランプリを獲得した「リアル・レンブラント」などはヨーロッパ方式の1,250本/50フィールドで制作され，そのままでは日本のハイビジョン受信機で見ることはできません。

ハイビジョン方式変換装置は，このような異なった方式を用いている国の間で番組を交換することを可能にしました(**写真 4**)。

しかし高精細なハイビジョンでは，現行方式よりも高精度な変換装置が必要で，これに必要な動き検出，動き補正などを高精度で行う信号処理技術の開発を進めています。

従来の方式変換装置では，動画像のボケやジャーキネス(動きが不自然で滑らかさがないこと)が問題となり，変換した画像は原画像に比べて著しく画質が劣るといった現象が生じていました。

今回展示した装置では，画像を小さなブロックに分け，おのおののブロックについて動きの情報(動きの方向と大きさ)を検出し，その情報をもとにして動き補正を行うことによって，画像の動きが滑らかでしかもボケの少ない高画質な変換を行うことができます。

高性能ハイビジョン 4板式CCD撮影実験

良質のハイビジョン番組を制作するには，高画質で機動性に優れ

〈写真５〉ハイビジョン４板式
CCDカメラによる撮像実験

〈写真６〉ハイビジョン領域抽出装置

た小型ハイビジョンカメラが必要です。

当研究所では，2/3インチ130万画素CCDを４枚使用することで，ほぼ260万画素CCDに相当する広帯域の輝度信号を得ることができ，かつ感度，ダイナミックレンジなども優れたハイビジョンカメラを試作しました(**写真５**)。

従来の３板式のカメラは，多くの場合プリズムを使ってレンズの入射光を赤，緑，青の光の成分に分解し，各成分に各１枚のCCDを使っています。CCDが３枚使われることから，３板式と言われています。

今回開発しました４板式のカメラは，レンズの入射光を赤，青，そして緑を２つに分解し，４枚のCCDを使用しています。このため，入射光を４つの光に分解するダイクロイックプリズムを新たに開発しました。

緑用の２枚のCCDは，互いに半画素ピッチずらしてダイクロイックプリズムにはり合わせています。おのおのの出力信号は合成され，広帯域の緑信号(48MHz)が得られます。

また赤，青の色信号についても，24MHzの映像帯域の信号が得られます。このカメラの水平限界解像度は約1,200TV本，感度は2,000ルクスF5.0，SN比は約52dBの特性が得られています。

ハイビジョン領域抽出装置
～HDムービーマット～

テレビドラマの列車内シーンで，事前に撮影した外の景色をはめ込んで合成する方法は，従来クロマキー手法としてよく使用されました。この場合ブルーバックという特殊な背景が必要で，通常の自然映像には使うことができません。

当研究所では，切り出したい対象を大まかに指定するだけで，任意の動画像から対象を追跡しながら自動的に切り出すことができるハイビジョン用領域抽出装置を開発しました(**写真６**)。

この装置では，切り出す境界線をオペレータが太い線で指定しますが，そこから先はコンピュータが境界線内で物体境界のエッジ(明るさや輝度が大きく変化する部分)を検出し，エッジをつないで真の境界線を求めます。この領域抽出装置を使用すれば，フルショットで撮影した走っている自動車からそのナンバプレートの部分だけ抜き取って数字を隠すこともできます。

MUSEの高画質化

ハイビジョンの映像は現行テレビの約５倍に相当する映像情報を含んでいます。このため現行放送と同じ衛星放送１チャネルで放送するためには，帯域圧縮をしなければ放送することができません。

毎日放送されているハイビジョン試験放送ではMUSE方式が使用されています。放送の受信画質は実用的に十分な品質ですが，MUSE方式は設計当初から将来の技術の進歩により，改良できることを配慮したものになっています。

〈写真7〉高画質化されたMUSEデコーダとその映像（左），および現行MUSEデコーダの映像（右）

〈写真8〉42GHz帯ハイビジョンディジタルFPU

〈写真9〉ハイビジョンシステム評価用標準動画像の例

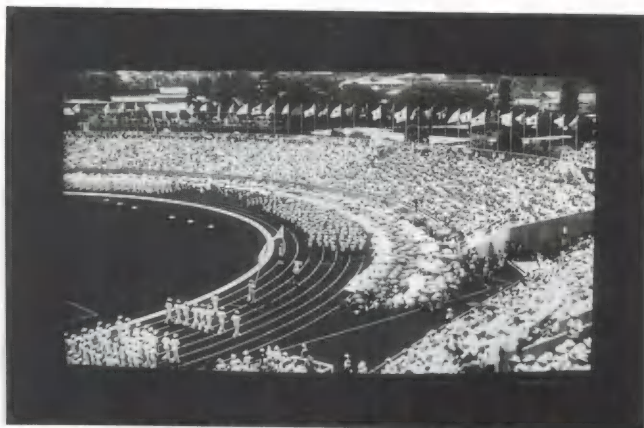
よりきれいな映像を放送できるよう，MUSE方式の高画質化の研究を進めています。

今回展示した装置の主な改善点は，動画像部分の精細度の向上，SN比の改善，リンギングの低減などです（**写真7**）。

42GHz帯ハイビジョンディジタルFPU

Jリーグや野球など，スポーツ番組をハイビジョンで生放送するときは，ハイビジョン映像の中継伝送装置（FPU）が必要です。42GHz帯を使用したアナログ方式の伝送装置は既に開発されていますが，降雨による電波の減衰が大きいため比較的近距離の伝送に限られていました。

そこで，ディジタル伝送技術を使用した高画質で伝送距離の長い42GHz帯ハイビジョンディジタルFPUを開発しました（**写真8**）。

今回展示した装置には，誤り訂正方式として送信部では高速の畳み込み符号，受信部ではビタビ複号を採用しています。

また，42GHz段を直接変調する方式や低雑音のPLLシンセサイザの採用などにより，伝送距離をアナログ方式の5kmから25km以上に拡大することができました。

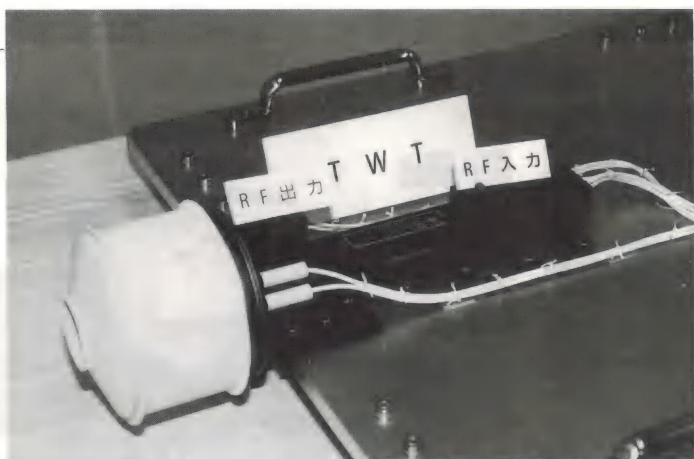
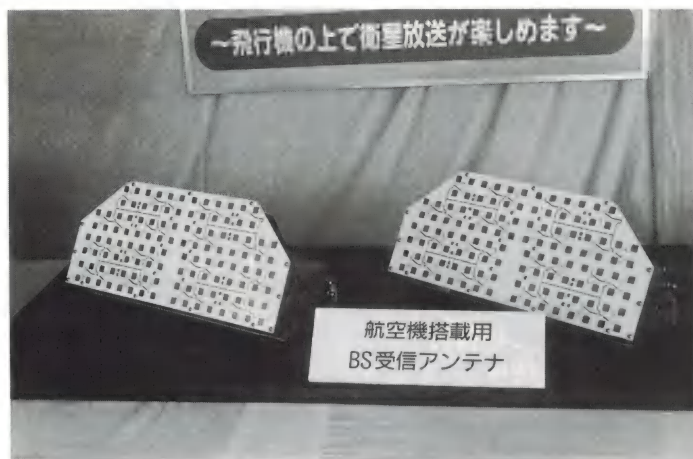
ハイビジョン・システム評価用標準動画像

最近，ディジタル圧縮技術が注目されています。ハイビジョンの信号をそのままディジタル信号に変換しますと，1.2Gbps（1秒間に12億ビット）の情報量になり，このままでは伝送したりVTRに記録したりするのが難しくなります。

この信号をできるだけ圧縮する技術が必要になりますが，圧縮を行えば画質に何らかの劣化が現れます。この劣化があまり目立たないようにするための研究を行っていますが，このようなシステムの総合的な評価をするためには，1つの画像を10～15秒間，評定者が見て判断することが一般的です。

画質の評価は，いろいろな絵柄の画像を用いて行いますが，普遍的な評価結果を得るためには共通の画像を用いる必要があります。

放送技術開発協議会（BTA）では，標準画像の選定を行っておりますが，当研究所でもハイビジョン画質評価の研究の一環としてその選定および制作に協力してきました。

今回の展示では画質の総合的な

〈写真10〉 12GHz帯 200W進行波管

〈写真11〉 航空機のレドームの中で動作する高能率台形平面アンテナ

評価に用いる標準動画像を紹介しました(**写真9**)。

将来の放送衛星の研究

●12GHz帯放送衛星搭載機器

現在，衛星放送はBS-3により120Wの出力で放送されていますが，小さな受信アンテナで安定に受信するには，高出力のBS送信装置が必要です。BS-3用120W TWT(進行波管)と重量で同程度まで小型化することができた高効率，高信頼性の200W TWTを開発しました(**写真10**)。

アンテナでは日本の主要都市で送信，受信とも高利得が得られる鏡面修整アンテナを開発しました。諸外国方向への電波の漏れが少なくなっています。

また，日本に割り当てられている8チャネルの送信電波を1つのアンテナで送信できるよう，合波するマルチプレクサも展示しました。挿入損失0.8dB以下で，200Wの信号を8チャネル合波することができます。

●21GHz帯放送衛星システム

21GHz帯(21.4~22GHz)の電波を使用すれば，広帯域の衛星放送ができますが，雨や大気の吸収による減衰が多くなりますので，現在の衛星放送とは別の対策が必要です。

日本を幾つものビームでカバーし，雨による地域ごとの電波減衰量に応じて，TWT出力の増減および合成を行って，ビームの送信電力を変化させる放射電力可変型システムを研究しています。

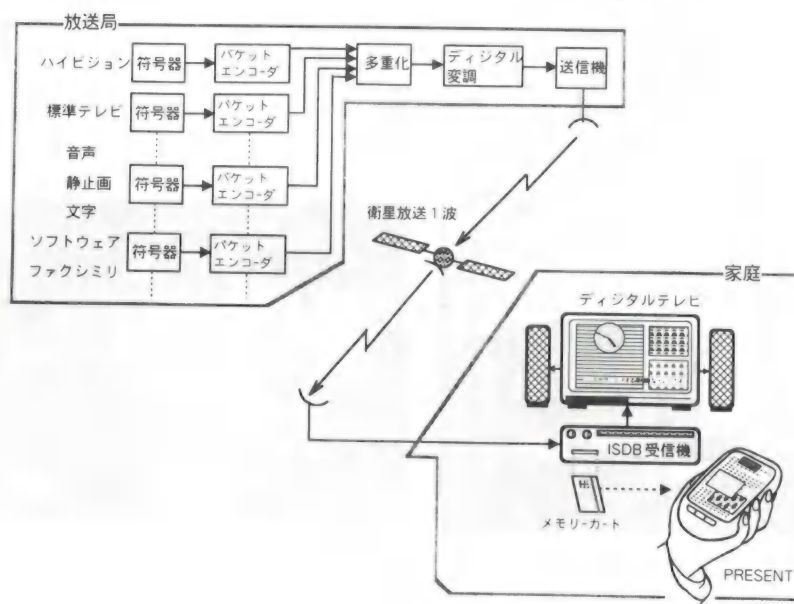
航空機搭載用BS移動受信装置

衛星放送は北海道から沖縄まで日本全国で受信できますから，航空機でも受信することが可能です。

衛星放送の特徴を生かすことができる，航空機搭載用BS移動受信装置の開発を進めています。この装置は高能率台形平面アンテナ(120素子，38cm×16cm)を2基使用し，航空機の胴体中央部に設置されたレドームの中で動作させます(**写真11**)。

アンテナを常に放送衛星に向けるため，航空機の航法装置から出力される機体の位置・姿勢データから衛星の方向を算出してアンテナの向きを制御します。この機械追尾と，受信信号の同相合成技術を用いた平面アンテナの電子追尾の両方式を使用して，衛星放送を受信します。

統合ディジタル放送-ISDB ~21世紀の放送マルチメディアサービス~

ISDB(Integrated Services Digital Broadcasting)は，今までの放送のように，1つのチャネルで映像と音声を送るだけのものではなく，1つのチャネル(伝送路)で多種類のサービスを混在させて放送することができる，21世紀を

【図1】
ISDBのシステム構成例

目指した新しいディジタル放送方式です。

この放送では，スタジオ並みの高画質ディジタルテレビや，カタログなどの文字情報をのせた静止画，コンピュータソフト，自動楽器の演奏情報を送るテレミュージック，ニュースや番組テキストのファクシミリなど，多数の情報を送ることができます(図1)。

このISDBでは新しい情報サービスとして文字情報や図形・静止画などで構成される高機能テレテキスト(PRESENT)の研究も進めています。これは受信機側で多様な番組検索ができるよう，番組表やキーワード表をインデックスデータとして放送し，必要な情報だけを選別することができます。

広帯域ISDB用 21GHz帯 変復調器

1992年に国際会議(WARC-92)が開催され，新しい衛星放送サービスのために21GHz帯の周波数が分配されました。この周波数帯では現行の12GHz帯に比べて，1チャネル当たりの帯域幅が広くとれるので，広帯域のISDBなどの伝送ができます。

今回の展示では，帯域幅約100MHzの広帯域衛星中継器を想定し，高速の誤り訂正方式として畳み込み符号を，変調方式としてQPSKを使用した変復調器を試作しました。

21GHz帯では雨や大気による減衰が大きいので，日本全国を複数のビームでサービスし，ビームごとに電力制御する衛星システムや，電力効率の良いディジタル変調方式などが必要となります。

地上ディジタル放送

FM放送を自動車で移動中に受信しますと音がとぎれて，良い音質で聴けないことがあります。これは電波のマルチパスフェージングによる受信信号の劣化が大きな原因です。

従来の伝送方式では受信側での完全な信号再生が難しいため，新しい伝送方式の研究を進めています。

ディジタル音声放送(DAB:Digital Audio Broadcasting)の伝送方式としては，複数のステレオ番組を広い帯域の中に多重し，OFDM(Orthogonal Frequency Division Multiplexing)変調する方式を検討しています。この方式を使用しますと，移動体向けにCD並みの高音質放送を低電力で実現することができます。

OFDM変調方式は，相互に直交関係にある多数の低ビットレート変調波を多数使い，受信側で受信波形をFFT(高速フーリエ変換)することにより復調する方式です。

マルチパスフェージングによって信号の一部が誤ったとしても，時間軸および周波数軸でのインタ

〈写真12〉 急激な受信劣化のない
AW-CDM方式のディジタル伝送

〈写真13〉 間欠スキャンアダプタ
を後部に装着したHARPカメラ

リーブと誤り訂正によって，完全な信号を再生することができます。

符号分割多重による階層的
ディジタル伝送方式
～急激な受信劣化のない
ディジタル放送を目指して～

ディジタル放送は，受信電界がある程度低下しても安定に受信できますが，強い雨や障害物で電界が下がったり，サービス限界点付近では，急激に受信品質が低下し，受信できなくなります。

今回展示したAW-CDM方式は，受信品質が受信状態に応じて緩やかに劣化していきます。低・中・高域に階層符号化されたテレビ信号を，低・中・高域情報の順に重み付けして符号分割多重(CDM)変調して送信します。

受信側では，CDM復調したあと優先度の低い高・中域情報の誤り率を測定し，高・中域情報が復元可能であればすべての情報を用いてテレビ信号を再生します。

受信状態が悪化し，高・中域情報の誤り率が，ある一定値を超え復元不可能になってきた場合は，受信状態に応じて高・中域情報を切り捨てて，誤りなく受信された情報の範囲でテレビ信号を再生します。この方式を使用しますと，従来のアナログ方式では良好な受信が困難な移動受信や，ポータブル受信が可能となります(**写真12**)。

現行放送改善のための研究

HARPカメラの
間欠走査による高感度撮影

HARP撮像管は，高感度で暗電流が極めて少ないという特徴があります。この影像管のビーム走査時間を間欠的にすることで，高感度の撮影を行うことができます。

テレビカメラは1秒間に60回，スチールカメラのフィルムに相当する部分を電子ビームで走査し映像信号を取り出しています。この走査をある時間たってから走査すると，その時間分，光の信号を蓄積することができ，通常の走査に比べて高感度の撮影が可能です。天体観測など，動きの少ない被写体の撮影に適しています。

今回，HARPカメラ(525本方式)に接続して使用できる間欠スキャンアダプタを開発しました(**写真13**)。これを使用して，間欠走査時間を最大8秒にしたとき，標準のサチコンカメラに比べて1万4000倍の高感度を得ることができました。この試作カメラ(モノクロ)を使用して，人間の目では見ることができないほど暗い(12等星以下の明るさ)放送衛星BS-3a, 3bの撮影に成功しました。

映像コンピュータ
HD Picot
～DTPPのための処理マシン～

HDはハイビジョン，PicotはPicture Computerの略で，プログ

ラムを入れ替えることにより，各種の映像効果を実現できるハイビジョン用リアルタイム映像処理装置です。

今回展示した装置には，処理スピードを上げるために映像処理専用に開発したプロセッサを64台組み込んでいます。これらを並行して動かすことにより，ハイビジョン信号を実時間処理することができます。プログラムのソフトにより，既存のDVEやカラーコレクタ，スイッチャの機能のほかいろいろな映像処理に利用することができます。

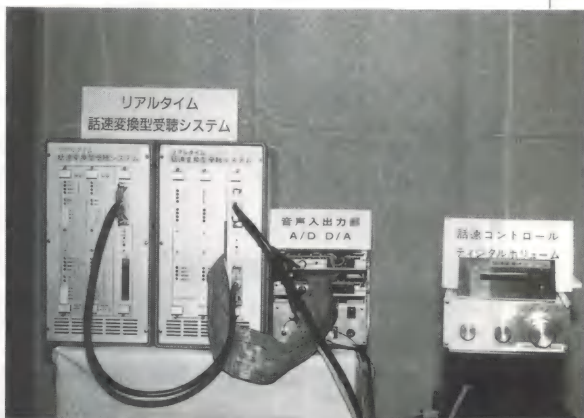
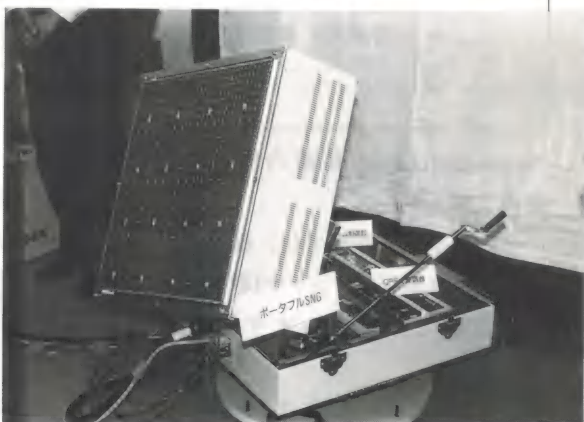
HD Picotを用いてワークステーション上で番組を作ることができるDTPP(デスクトッププログラムプロダクション)をすでに開発しましたが，この装置を使用しますと，コンピュータグラフィックスの映像と，カメラで撮影した映像を合成して表現することも可能です(**写真14**)。

〈写真14〉
DTPPのための
処理エンジン HD Picot

〈写真15〉
リアルタイム話速
変換型受聴システム

〈写真16〉
ポータブルSNG装置

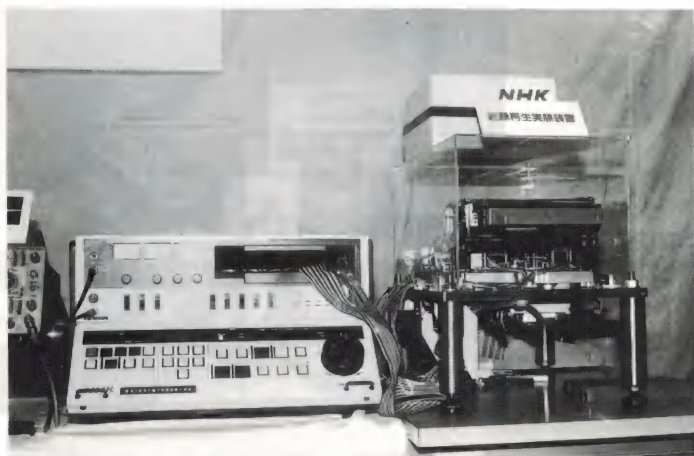
リアルタイム話速変換型受聴システム

21世紀の長寿時代に備えて，お年寄りにも放送を楽しんでいただくために，早口で話された音声を，「ゆっくり」した音声にリアルタイムで変換できる話速変換システムを開発しました(**写真15**)。

この装置では，入力音声を有声，無声，無音区間に分割して音声を処理し，話しはじめを遅く，終わりを速くすることにより，声の高さや質(個人性)をほとんど変えずに聴感上の話す速度を自由に変えることができます。このシステムは話す速度を遅くしても内容時間が変化せず，音声の品質劣化が極めて小さいことが特徴です。

平面アンテナを用いたポータブルSNG装置 ～コンパクト化と自動追尾による取材の機動化～

中継車の入れない山奥など交通の便が悪い場所で，事故や災害が発生し，緊急にニュースの報道が必要になることがあります。

当研究所では，このような緊急報道時に，通信衛星を使用して伝送できる，小型で軽量なポータブルディジタルSNG(Satellite

〈写真17〉 メガネなし立体ハイビジョン

〈写真18〉 高密度垂直磁気テープの記録再生実験

News Gathering)装置の開発を進めています。

今回展示した装置では，60cm角の送受信共用平面アンテナ(利得：37dBi)を使用しています(**写真16**)。

アンテナを衛星に向ける際，衛星からのビーコン電波により衛星方向を検知し，自動追尾していますので細かい方向の調整が不要なため，素早いセッティングが可能です。

平面アンテナの裏面に一体化された固体化電力増幅器(SSPA)により送信eirp(等価等方放射電力)は54dBW得られます。アンテナおよび送信機全体が1個のトランクに収納(総重量は40kg)できるので運用性が飛躍的に向上しました。

将来の放送を支える研究

メガネなし立体ハイビジョン

21世紀の放送サービスとして，奥行きを表現できる立体テレビシステムの基礎研究を進めていますが，今回は，特別なメガネを用いなくても見ることができる70型液晶投写式立体ハイビジョンを展示しました(**写真17**)。

立体表示の原理は，右目用と左目用の別々の映像信号を合成して，高解像度液晶ビデオプロジェクタにより対角長180cmのレンチキュラスクリーンに背面投写する方式です。

レンチキュラスクリーンのレンズ作用により，約3m離れた位置では左右の目に映像が分離されて入るため，メガネをかけなくても高画質で臨場感のあるハイビジョン立体映像を見ることができます。

高密度垂直磁気テープの記録再生 ～HD VTRの小型化，長時間化を目指して～

ハイビジョンの映像信号をディジタルで小型カセットテープに記録するには，超高密度記録が必要です。

現在，ほとんどのVTRで用いられているテープ長手方向の磁気記録方式は，記録周波数が高くなるほど信号の記録が難しくなります。

一方，垂直磁気記録方式(磁性面に垂直に磁化)は，逆に記録周波数がある程度高い方が安定な記録ができます。この垂直磁気記録方式に用いる高性能な垂直磁気テープを開発しました。

今回展示したテープは，厚さ10μmのベースフィルム上に，記録層として従来のコバルト・クロム組成に微量のタンタルを添加することにより，良好な性能を得ることができました(**写真18**)。

光るシリコン ～多孔質シリコンの電子線励起発光～

シリコン結晶は，トランジスタやICによく使われる廉価な材料で，これまで発光材料にならないとされていました。

しかし最近，そのシリコン結晶に特殊な加工を施し多孔質化(軽石のような穴だらけの構造)すると，発光することが発見され，新しい発光材料として注目されています。

多孔質層に電子線を照射すると発光し，電子線の加速電圧により白色あるいは赤橙色に発光します。当研究所では，この発光機構の解明とデバイスへの応用を目指して，多孔質シリコンの作製と発光に関する研究を進めています。

今回の展示では，電子線照射による発光実験を紹介しました。

立体テレビのための音像遠近制御

音声放送は，モノラルからステレオへ，さらにハイビジョンの3-1方式へと開発が進められてきました。21世紀の立体テレビに対応して音像も3次元的に飛び出させることができれば，迫力のある音を楽しむことができます。

今回は，ディジタルディレイ装置を用いて，多数の小スピーカから出る音波のタイミング(位相)を調整するシステムを展示しました。

空間の1点に音圧を合成して，音像を手前に出させたり，また，遅延時間とレベルを外部から制御することで，焦点の位置を連続的に移動させることができます。このシステムの特徴は，音像を飛び出させることができるだけでなく，比較的広い範囲で飛び出す音像を聴取できることです(**写真19**)。

〈写真19〉 音像遠近制御の展示

〈写真20〉 移動受信用FM多重放送　音声信号のすきまを利用してディジタル信号を多重し，走行中の自動車に交通情報など文字や図形を伝送します

おわりに

技研公開の中から主な項目を紹介しましたが，これ以外にも興味深い展示が多数行われました。今年の公開は昨年より公開日を1日増やして3日間とし，多くの方々に研究の成果を見ていただくことができました。

今年はテレビ放送40周年にあたり展示，講演会，研究発表会と多彩な企画をしました。5月28日には「マルチメディア情報産業を変革する力」と題して米国ベルコア社のアーウィン・ドロス博士の講演会を行い，大変好評でした。

また，5月29日には若手研究者による9件の研究発表が行われました。活発な質疑応答が行われ，特にディジタル放送関連の発表に強い関心がよせられました。

今回展示された技術すべてが即，明日の放送に役立つものとは限りませんが，21世紀を目指した新しい放送技術として，これらの研究成果を皆さまの茶の間にお届けできる日が来ると思います。

当研究所では，さらにより良い番組を放送するために，ディジタル放送や将来の放送を目指した技術の研究に努力が続けられています。

(NHK放送技術研究所　総務部)

スキップバックレコーダ(SBR)の開発(NHK)

NHK大阪放送局で，事件･事故などの決定的瞬間を確実に記録するSBR(Skip Back Recorder：遅延映像収録装置)が開発されました。

これまで，地震発生の瞬間や野球のホームランシーン，サッカーのゴールシーンのような予測できない出来事は，VTRの長時間収録や高価なディスク装置などによりその瞬間の映像が記録されていました。

今回，映像遅延メモリーを使用することにより，出来事が発生した時点で過去にさかのぼってその決定的瞬間を収録する装置が開発されました(**写真，図**)。

本装置は低廉で，安定した運用性と簡単な操作性を有しており，地震などの報道取材やスポーツ中継だけでなく，各種センサ(動き，音声，赤外線など)と組み合わせることにより，動物の生態記録などのドキュメンタリーなどにも応用できるほか，防犯システムや障害探索システムなど放送以外の分野への応用も期待されます。

NHKでは，去る6月19日のプロ野球中継から使用され，その後，報道取材や各種番組で活用されています。

なお本装置は，エレテックス㈱の協力で製作されたものです。

SBRの概要

カメラの映像信号は，本装置の映像遅延メモリー(5秒に設定)を通すことにより，常に5秒遅れた映像がVTRに入力されています。出来事の発生を確認した時点でVTRを起動すれば，過去にさかのぼってその決定的瞬間を収録することができます。

本装置は低廉化を図るため，特殊再生機能(スロー，コマ送り，フリーズなど)は省き，5秒間の動画遅延機能のみに限定されています。

今後，カメラへの組み込みなど多方面への利用が考えられています。

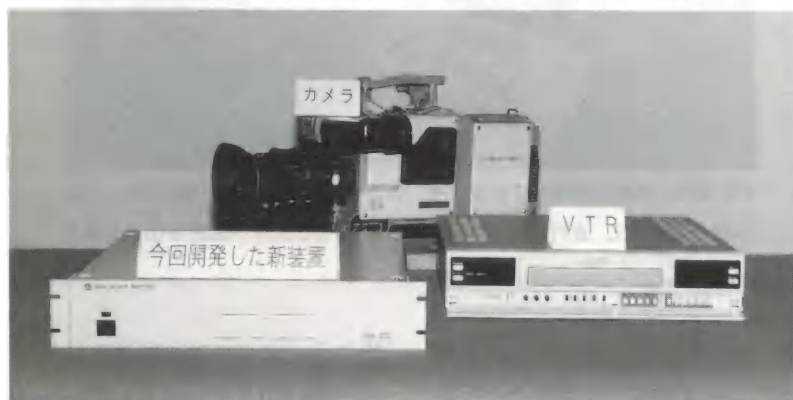
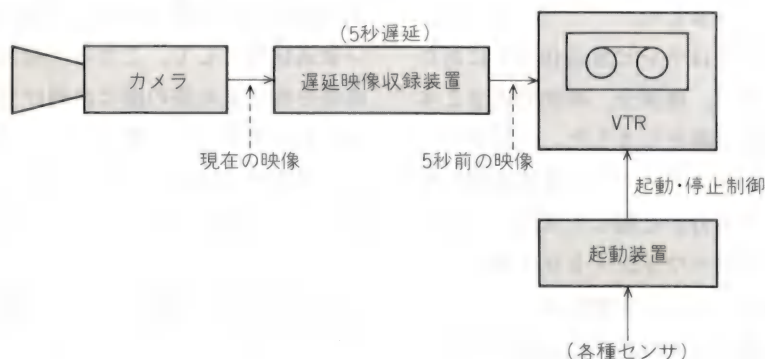
諸元

○入出力信号
　NTSCコンポジット信号(出力信号は2系統)

○映像遅延メモリー
- 映像遅延時間；約5.1秒(154フレーム)
- サンプリング周波数；14.31818MHz(4fsc)
- 分解能　；8bit(直線)

○電源
- 使用電源；AC100V50/60Hz
- 消費電力；40W以下

○外形
- 外形寸法；W480×H88×D400㎜
- 重量　；約9.5kg

(編集)

〈写真〉 SBRシステムの外観

【図】 SBRシステム

皇太子ご結婚中継放送制作記

竹之下　清治

6月9日に結婚の儀を終えられた皇太子ご夫妻のパレードは，雨上がりの砂利に，みずみずしい初夏の緑葉が茂る皇居東庭からスタートしました(**写真1**)。喜びにあふれ，晴れやかな皇太子殿下と雅子妃の表情を放送でご覧になった方も多いと思います。

NHKでは34年ぶりに行われた皇太子ご結婚の儀を，現在のテレビで放送と同時にハイビジョンでも放送・記録しましたのでその概要を紹介します。

〈写真1〉
皇居前広場

〈写真2〉
小和田家前

当日の動きと中継放送の特徴

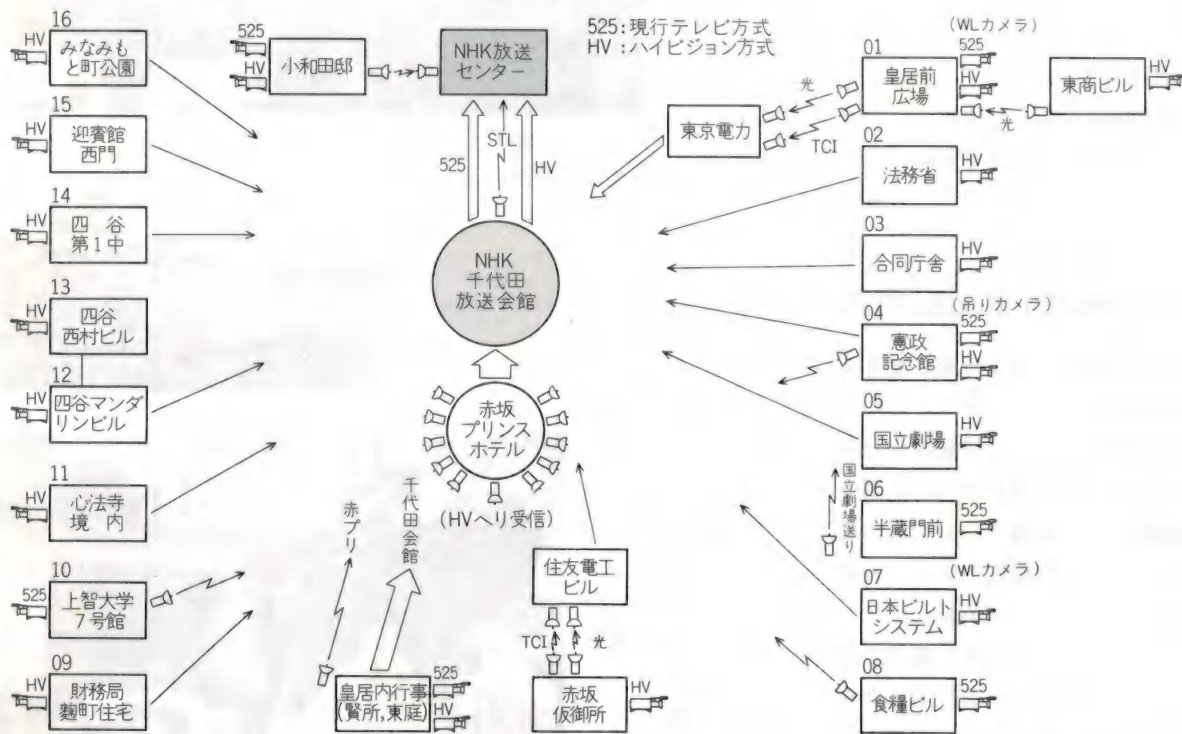
当日の取材は朝5時過ぎ東宮侍従の使者出発から始まりました。6時過ぎ小和田家では使者のお迎えを受け，雅子様の出発，玄関前(**写真2**)のご家族とのお別れの挨拶，そして皇居桜田門から2重橋門へと行事に沿って展開しました。

皇居内では結婚の儀，朝見の儀など多くの儀式があり，沿道から祝福を受けられるパレードでお祝いムードは最高潮となりました。

NHKではこれらの行事や儀式の中継を，サイマルキャスト方式(現在の放送とハイビジョン放送を1台のカメラで利用する方式)で実施しました。

サイマルキャスト方式は2つの方式の両立をめざしたものです。この方式は，大相撲や甲子園の高校野球中継などで実験が重ねられていますが，今回は，皇居賢所や歓迎の雑踏の中など放送実施にあたり取材場所の制限もあり，機材の有効活用を考えてこの方式を採用しました。皇居をはじめ東宮仮御所，パレードなどこの方式による広範囲で大規模な中継は初めてのことです。

この中継にはハイビジョンのヘリコプタ中継や，120mにわたる吊

【図１】パレードコース中継配置図

【図２】皇太子結婚の儀・パレード中継概念図(サイマルキャスト方式)

り移動撮影カメラシステムも開発して使用しました。

皇居前広場での撮影には，NHK放送技術研究所で開発したハイビジョンワイヤレスカメラ(WLカメラ)を初めて使用しました。ケーブルの道路横断が不可能なときやフィールドなど近距離で動きのある撮影に有効な手段となります。

なお，音声はステレオで放送しました。

中継放送

〈中継の規模〉

- スイッチングセンター１個所
- マイクロ受信基地４個所
- ハイビジョンカメラ54台
- 中継車23台
- ハイビジョン用ヘリコプタ１機
- ワイヤレス無線カメラ５台

などを使用しました(**図１，２**)。

このほか現行テレビ方式カメラとハイビジョンカメラとの混在使用のためダウンコンバータを使用しています。

〈皇居内の中継〉

結婚の儀が行われた宮中賢所にはハイビジョンカメラ3台，参列者の入場を撮影する参集所の1台，の計4台を1台の中継車でまとめました。

〈写真3〉
皇居東庭前のカメラ

宮内庁でのご両親の会見や朝見の儀，パレードのためにハイビジョンクルーと6台のカメラを使用(**写真3**)，中継車でハイビジョンの収録や現行方式に変換するダウンコンバートも実施しました。

計2台のハイビジョン中継車からは，光ファイバ回線でNHK千代田放送会館のスイッチングセンターへ伝送しました。

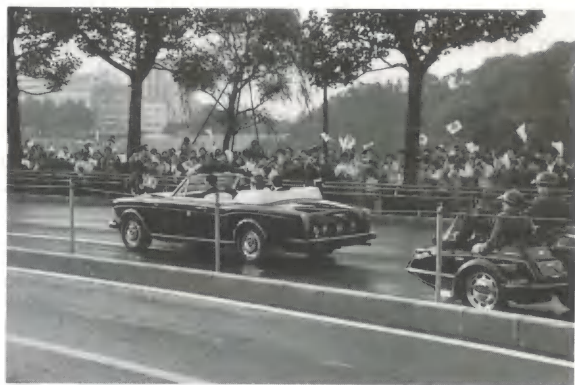
木立に囲まれた静寂の中で行われた儀式の雰囲気を十分に伝えることができました。

〈パレード中継〉

NHK放送センター内では，現行テレビ方式のスタジオとハイビジョンスタジオを使用しました。

各所に散在する中継場所(**写真4，5，6，7**)で取材した映像・音声は，マイクロ回線や光ファイバ回線によりスイッチングセンターに集めました。

スイッチングセンターでは現行方式のスイッチングとハイビジョン方式のスイッチング機能を備え，音声・ミキサ機材もそれぞれ分離して仮設しました。同じ出先現場を2つの方式で利用するため，連絡系統が非常に複雑な構成となりました。

〈写真4〉パレード中継

〈写真5〉パレード

〈写真6〉パレード中継

〈写真7〉パレード中継

〈写真8〉スイッチングセンター

ハイビジョンの伝送は光ファイバ回線を利用しました。電話回線と同じように電柱から架空したりマンホールから回線を割り出して中継車まで配線しました。

光ファイバ回線の布線が不可能な場所は，TCI(Time Compressed Integration)コーダによる42GHzのハイビジョンFPUとディジタルの光FPUを使用して本線，予備回線として伝送しました。

TCIとは，ハイビジョンの輝度信号と色差信号を時間圧縮して1走査線区間に時間分割多重するものです。

音声は，垂直走査期間にPCM化して重畳しています。

16台の中継車から送られた映像・音声はスイッチングセンターで現行・ハイビジョンともに1本の素材として制作し，光ファイバ回線で放送センターの現行方式とハイビジョンの2つのスタジオへ送り，完プロとして放送しました。

放送のほかに同時に代表取材・プール映像音声を制作し関係各放送局へ分配するとともに，海外放送機関へも提供しました。

〈スイッチングセンター(S.W.C)〉

中継場所が多数で広範囲にわたるため，今回パレードコースに近い千代田放送会館にS.W.Cを仮設しました。通常番組を制作するスタジオでは，今回のような大規模な中継では対応できず機能が不足するため機材を持ち込み仮設しました(**写真8**)。

使用目的に合わせ系統を整理し運用しやすいように，可搬機材で大規模な副調設備を構成しました。いつもは公開番組などに使用している250㎡のホールを現行・ハイビジョン用と2つに分け，両方で共用する機器はお互い見易い位置に配置しました。

各現場からの映像は，調整のうえスイッチャで選択され送出モニタで最終的にチェックされたのち，光ファイバ回線で放送センタースタジオへ出力しました。

パレードの音声は雰囲気を大切にしたステレオ放送です。映像系と同じように各出先集音場所から集めた音声を整音してS.W.Cでミキシングしました。ダウンコンバータによる音のズレ補正などの処理も同時に実施しました。

音声もクリーンサウンドをベースに両方式の放送としました。

新しい試み

①ハイビジョン・ヘリ中継

現行テレビ方式のヘリ中継技術は，確立されており毎日の放送に利用されています。

しかし，ハイビジョンは衛星による実験放送でもおわかりのようにNHK放送技術研究所で開発された帯域圧縮方式「MUSE」を使用しても帯域幅27MHz必要ですが，最近のディジタル圧縮技術の発達で帯域の高圧縮が可能となってきています。

今回実験的に，現在利用しているヘリシステム技術と圧縮技術に特殊変調方式を組み合わせ，現行のB・Cバンドによるハイビジョン伝送システムを開発し使用しました。

受信アンテナの自動追尾は簡易型を使用，伝送距離が数kmでも，非常に良好な結果が得られました(**写真9**)。

②移動吊りカメラシステム

クレーンカメラやポールカメラなど立体的な撮影を試みましたが警備や安全上の理由で実現できませんでした。その中で唯一使用で

きたのがこのシステムです。

お車が二重橋を出て間もない皇居前広場の内堀通りで，松の枝越しに移動しながら撮影された映像をご記憶の方も多いと思います。パレードコースと並行に張ったワイヤに小型カメラを吊るして移動撮影(120m)したものです(**写真10**)。

このシステムの原型は，人出の多い「博多どんたく」祭りの撮影に考案されたもので電柱などを利用します。時速7～8kmで進行する車列を追うこの撮影方法は，単調になりがちなパレード中継のアクセントとしてキラリと光る映像となり好評を得ました。

おわりに

現天皇陛下のご結婚によって白黒ながらテレビが家庭に急激に普及したといわれます。34年ぶりの皇太子殿下のご結婚中継は，新しい時代にふさわしいハイビジョンによって撮影し放送と記録することができました。

朝から降り続いた雨も，国民の直接の祝福を受けられるパレード直前にはやみ，計画どおりオープンカーによるパレードが実施され晴れやかな表情のお二人を放送することができました。

ハイビジョン受信機は100万円以下の機種も発売され，新しい時代の幕開けにふさわしい中継放送ができました(**写真11**)。現行方式の放送もハイビジョン機器の使用により高画質の放送となりました。

今回の中継実施にあたっては，郵政省はじめ多くの放送機器メーカーとNTTやTTN，NTS，JHV各社など多くの方々にご協力をいただきました。誌上を拝借して感謝の意を述べさせていただきます。

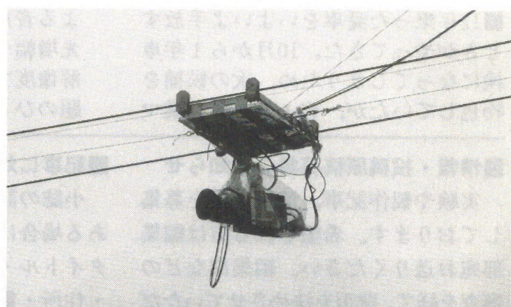
(NHK放送技術局
報道技術センター　中継制作)

〈写真9〉
ハイビジョン
中継用ヘリコプタ

〈写真10〉
移動吊りカメラ

〈写真11〉ハイビジョン放送公開

10月号特集予告

9月17日発売

特集＝【実験で学ぶディジタル回路】(仮)

基本のロジック回路の動作を理解すると同時に，回路をいろいろと応用できるように，
教科書的ではなく，実験を通してわかりやすく解説する。

①ゲート回路　②シフトレジスタ回路　③コンパレータ回路　④カウンタ回路　⑤応用例

◆

連載記事「実験で学ぶエレクトロニクス」「これでわかるOPアンプマスター講座」
「違いがわかる98 VS マック」は，9月号で終了いたします。
なお，10月号より新連載を予定しております。

編集後記

■7月の人事異動で「EL誌」は，「趣味文芸部」で行うことになりました。それに伴って，雑誌と同時に単行本の企画・編集も進めることになりました。そこで皆さんにお願いです。雑誌の投稿募集ももちろんですが，エレキに関係なくても道楽・ホビー・趣味の分野で単行本(ムックも含む)向きのネタ・アイディア・人物を募集します。我はと思われる方・こんな人がいる，などの情報をお待ちしています。Fこと藤本へ　〈F〉

■12年乗った愛車をいよいよ手放すときがやってきた。10月から1年車検になってしまうため，次の候補を物色していたが，ディラーの営業マンの熱意に押し切られて決めた。実は，購入車が8月にモデルチェンジするために，いま購入すると値引きが良かったところが本音だ。条件はツインカムエンジン，オートマチック，4ドアで，当方の購入額を提示して，それに見合う在庫があったから。〈H〉

■AV機器と人とのインタフェースは，振動板スピーカとCRTが今も主流で，原理的にも画期的なものはまだ少ない。前号で紹介された〝摩擦による音声の発生〟や，今月号の〝空間光増幅デバイス〟を使った高輝度・高解像度プロジェクタなど，新しい話題のひとつでしょうか？小型で良質の超低音スーパーウーファや，薄型・大画面の液晶やプラズマディスプレイなどがロープライスで家庭に浸透する日を楽しみにしたい。〈T.U〉

■次々と出てくる魅力的なパソコンのハード／ソフト，何をどう使うかこの選択が難しいところ。このような中でパソコンを使ってみたいが，面倒で難しくて，という人たちのために設計されたというMacintoshについて，このMacに関心を持っている人，あるいはMacを初めて使いたいという人たちを対象に，Macに関する基本的事項やそのときどきの話題を取り上げ「Mac ア・ラ・カルト」として連載を予定しています。〈I〉

■情報・投稿原稿募集のお知らせ

実験や製作記事，情報などを募集しております。希望される方は編集部宛お送りください。編集部などの審査を経て，採否を決めさせていただきます。採用された原稿等は規定の原稿料をお支払いいたします。詳細は編集部までお問い合わせください。

■記事に対するご質問は

小誌の記事に関するご質問などがある場合は，何年・何月号・何頁・タイトル・筆者名・質問内容・氏名・住所・電話番号などを明記のうえ，切手を貼った返信用封筒を同封するか，往復ハガキで編集部宛にお送りください。

●バックナンバーと年間予約のお知らせ●

書店にない場合や，バックナンバー，年間講読をご希望の際は，左記に示す当社「電話注文センター」までお申し込みください。なお，年間講読料は9,000円(送料込み)です。

また，ハガキでの注文もできます。

EL エレクトロニクスライフ

1993年9月号 通巻733号
1993年9月1日発行
定価750円(本体728円) 〒76
編集・発行 日本放送出版協会
〒150 東京都渋谷区宇田川町41-1
☎03-3464-7311(代表)
印刷：大熊整美堂　製本：石津製本

直接注文は「電話注文センター」へ
電話注文センター
03-3780-3339
(月～金，午前9:30～午後5:30)
振替　東京1-49701

日本放送出版協会 〒150 東京都渋谷区宇田川町41−1 ☎(03)3464−7311

定価七五〇円（本体七二八円）